





Licence MIAGe

Année 2010-2011

RAPPORT DE STAGE

Outil pour la Traçabilité entre Exigences et Use Cases

Stage de 5 mois effectué au sein de

la Société Airbus Operation SAS par :

PHAN Ngoc-thang

Mémoire soutenu à l'Université le 05-09-2011

Encadrant : D. Michelet	Tuteur : Jean BOUSQUET

SOMMAIRE




AIRBUS

TABLE DES ILLUSTRATIONS




AIRBUS

REMERCIEMENTS

Je tiens tout d'abord à remercier mes parents, Monsieur **PHAN Ngoc-Tu** et Madame **TRAN Thi-Tam-Tuyen**, de m'avoir permis de réaliser les études à l'étranger, de supporter mes décisions, de m'encourager quand j'ai des soucis, de m'écouter et de me donner des conseils quand j'en ai besoin. Merci pour leur soutien apporté, car sans eux, je ne pourrais pas arriver jusqu'à aujourd'hui.

Je tiens aussi à remercier mon tuteur de stage chez Airbus, Monsieur **BOUSQUET Jean**, de m'avoir accepté au sein d'Airbus. Merci pour son suivi tout au long de mon stage, pour sa gentillesse et sa disponibilité. Je remercie aussi Madame **SIGNES Nathalie**, avec qui je travaillais la méthodologie V&V, de m'avoir beaucoup aidé dans la prise de connaissance des nouvelles méthodes, dans la rédaction du rapport de stage et des autres documents techniques. Je remercie également les autres membres de l'équipe SEVS (M **GUEHL Patrick**, M **LAMOTTE Jean-Louis**, M **BARRERA-ESQUINAS Julian**, M **GIGOT Christian**, Mme **LAMAILLOUX Françoise**, Mme **DUSCH Elisabeth**) de m'avoir consacré leur temps pour mieux m'intégrer dans le service, merci à M **ARTIGUE Yves** et tous les autres membre de SEVO, SEVC de m'apporter une bonne ambiance de travail avec leur bonne humeur. Merci à M **PHAM Quang-Vinh** du service de Technical Data, de m'accompagner durant mon stage à Airbus.

Au sein de l'université, je tiens à remercier M **MICHELET David**, mon tuteur de stage côté université, de m'avoir suivi dans mon stage, de m'avoir conseillé pour le déroulement de mon stage. Je remercie aussi tous les enseignants de la MIAGe pour m'avoir permis une réelle formation de gestion informatique.

Je remercie aussi M **ZUFLUH Giles**, directeur de la MIAGe, Mme **EMERITT Fabienne**, secrétaire de la MIAGe, M **LE Luong** de me donner les conseils dans la recherche de stage et M et Mme **POGNON** de leurs aides et leurs conseils dans mon rapport de stage.

Enfin, je souhaite envoyer un grand merci pour tous mes amis qui m'ont supporté durant le stage.



PRÉSENTATION DE L'ENTREPRISE

EADS

European Aeronautic Defence and Space company (*EADS*) est un groupe industriel du secteur de l'industrie aéronautique et spatiale civile et militaire. Il est l'un des premiers groupes de défense en Europe et dans le monde. EADS est une société qui était créé le 10 juillet 2000 par la fusion de trois sociétés Aérospatiale Matra (AS Matra) (France), Construcciones Aeronáuticas SA (CASA) (Espagne) et Daimler Chrysler Aerospace (Allemagne). Il est constitué de cinq divisions :

- AIRBUS pour les avions de transport civil et les avions militaires: 62% CA total (22 179 M€), 47% d'effectif (54721)
- ATM Avions de Transport Militaire : 2% CA total (763 M€), 7% d'effectif (7536)
- Eurocopter pour les hélicoptères civils et militaires : 9% CA total (3211 M€), 11% d'effectif (12755)
- EADS Astrium pour les satellites, les lanceurs civils et militaires : 8% CA total (2698 M€), 10% d'effectif (10985).
- Systèmes de Défense et Sécurité pour les avions de combat militaires (EUROFIGHTER), les missiles et autres systèmes militaires 16% CA total (5636 M€), 21% d'effectif (23237)

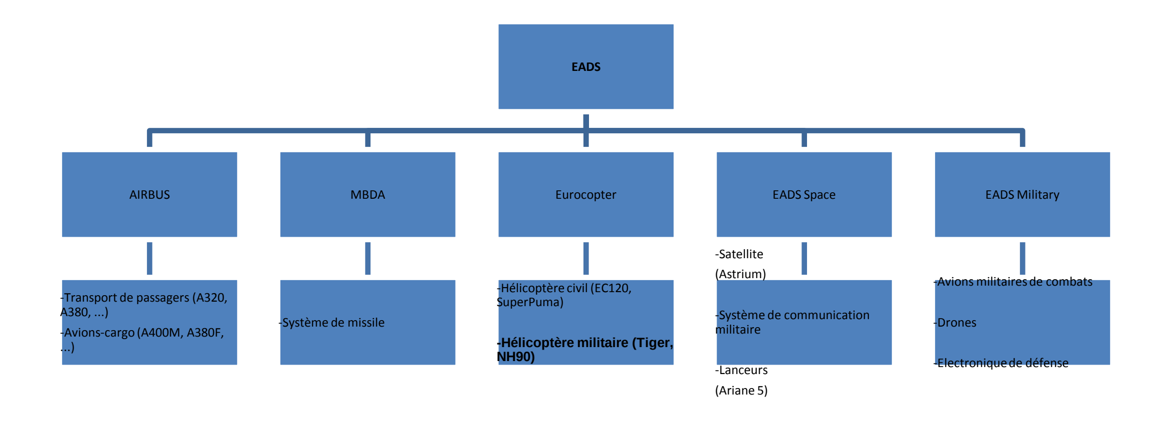


Figure 1 - Le groupe EADS



Airbus Operations SAS

Organisation

Basé à Toulouse, France, Airbus est une société EADS, constituée en vertu de la loi française comme une société par actions simplifiée ou «SAS» (Société par Actions Simplifiée). La conception d'Airbus et de sites de production sont regroupés en quatre filiales en propriété exclusive, Airbus Operations SAS, Airbus Uperations GmbH, Airbus Operations SL et Airbus Operations Ltd. Un comité exécutif de dix membres, dirigé par le CEO et président d'Airbus Tom Enders, et notamment les chefs de chacune des fonctions de métier de la société, est chargé de gérer toutes les activités d'Airbus.

Airbus est le meilleur constructeur aéronautique européen qui capte environ la moitié des commandes d'avions mondiales, avec 52 500 employés de 88 nationalités dans le monde, implantés en France, en Allemagne, en Espagne, au Royaume-Uni, en Amérique du Nord, en Chine, en Inde, au Japon et en Russie et un réseau mondial comptant plus de 327 clients et 348 opérateurs.

En 2010, Airbus a réalisé un chiffre d'affaires de près de 30 milliards d'euros (avec un part de marché brute (unités) de 50.7 %) et a apporté son soutien à l'opération très fiable de plus de 6.500 avions d'Airbus actuellement en service avec plus de 300 opérateurs dans le monde. Jusqu'à aujourd'hui, Airbus a reçu plus de 10 060 commandes d'environ 323 clients. Pendant les 6 premiers mois de l'année 2011, avec le succès de l'introduction de l'A320neo, Airbus a reçu 777 commandes et effectué la livraison pour 358 avions aux clients.

Histoire

Création



Airbus a été créé le 18 décembre 1970 comme un consortium de constructeurs aéronautiques européens, sous forme d'un groupement d'intérêt économique entre l'Aérospatiale (France) et la Deutsche Airbus (Allemagne) en réalisant que seulement par coopération la construction aéronautique européenne pourrait concurrencer efficacement les américains. En 2001, 30 ans après sa création, Airbus est officiellement devenu une société unique et intégrée, passant ainsi une autre étape importante dans son histoire : la Société EADS et BAE System du Royaume-Uni sont devenu actionnaires d'Airbus avec 80% et 20% pour chacun.

Les premiers partenaires



Les deux partenaires à part entière dans le consortium d'origine ont été Aérospatiale pour la France et Deutsche Aerospace pour l'Allemagne. Hawker Siddeley et Fokker ont également été associés au programme et l'espagnol CASA est devenu un membre à part entière du GIE en 1971. Initialement basée à Paris, le GIE déménage à Toulouse en 1974. British Aerospace est devenu un partenaire à part entière en 1979.

Le premier avion d'Airbus, l'A300B, a été lancé lors du salon de Paris 1969. Il a été le premier gros-porteur biréacteur et pouvait transporter 226 passagers dans un aménagement confortable à deux classes lay-



out. Une version de 250 sièges, l'A300B2, demandée par son client de lancement, Air France, est entré en production à pleine échelle.



En 1974, l'A300 a été certifié selon le budget et en avance sur le calendrier - une grande première pour les entreprises européennes à l'époque. À la fin de 1975,Airbus avait 10% de part de marché et un total de 55 appareils en commande. L'entreprise a alors connu une période sombre, durant laquelle il n'a pas obtenu de nouvelles commandes. Enfin, la compagnie aérienne américaine Eastern Airlines a décidé de louer quatre A300B4s.



C'était un tournant, et à partir de là, Airbus n'a jamais regardé en arrière. Dans les deux ans, Airbus avait 133 commandes fermes et sa part de marché avait augmenté à 26 pour cent en valeur. À la fin de 1979, Airbus avait 256 commandes de 32 clients et 81 avions en service auprès de 14 opérateurs.

Pousser la famille



En Juillet 1978, Airbus a lancé l'A310, une version raccourcie de l'A300 avec une capacité de 218 passagers. Airbus n'est plus constructeur d'un unique avion. Après cette course, British Aerospace - qui avait repris Hawker-Siddeley - est devenu un partenaire à part entière du consortium Airbus en 1979. Tous les grands constructeurs européens sont désormais solidement unis et prêt à défier l'industrie américaine.

Cette même année, Airbus a décidé d'intégrer un avion de 130 à 170 places à couloir unique, sur lequel les partenaires ont travaillé en dehors du consortium, dans sa famille d'avions. Ce projet est devenu l'A320, qui remplit la ligne de produits d'Airbus et a permis à Airbus de concurrencer l'industrie américaine pour les remplacements des avions vieillissant de cette catégorie, en service dans le monde entier à l'époque.

Présenter l'A320

L'A320, lancé en 1984, était le premier tout nouveau design de sa catégorie en 30 ans. Intégrant les nouvelles technologies, l'avion fourni une meilleure efficacité de fonctionnement, une meilleure performance et - surtout - un plus grand confort des passagers. Il a établi la norme pour tous les cockpits Airbus ultérieurs et même pour l'industrie dans son ensemble.



L'introduction de fly-by-wire a également permis à Airbus de développer une famille d'avions partageant le même cockpit et les mêmes caractéristiques de conduite de vol.

L'A320 a été suivi en 1989 par l'A321, une version allongée, embarquant 185 passagers dans un configuration standard à trois classes, et, en 1992, par une version de 124 sièges - l'A319. La famille monocouloir a été achevée en 1999 avec l'introduction de l'A318 de 107 sièges.

La décision de lancer l'A320 a rencontré un gros succès. En dépit de la récession du milieu des années 80, la demande du marché s'est retournée en faveur d'un avion moderne « cost-efficient » pour remplacer les avions anciens de même capacité. Le nouvel A320 a été rapidement choisi par Air France, British Caledonian, Adria Airways, Air Inter et Cyprus Airways. Aujourd'hui, il est l'un des best-seller des avions Airbus et très populaire auprès des passagers et des transporteurs.



LANCER LA FAMILLE DE LONG-RANGE

En 1987, il était clair pour Airbus que le moment était venu de lancer non pas un, mais deux gros avions dans un seul programme. Le marché était prêt pour les avions « moyen-courrier » ainsi que d'une longue portée, quatre moteurs. Les deux nouveaux avions de ligne ont partagé la même cellule, le design des ailes et la même section bi-couloir que l'A300/A310, intégrant le concept fly-by-wire de l'A320.



Lorsque le quadriréacteur A340 est entré en service en 1993, il a été le premier avion long-courrier entièrement nouveau en opérations commerciales depuis plus de 20 ans. Le biréacteur A330 qui l'a rejoint un an plus tard combinait le plus bas des coûts d'exploitation de tous les avions avec une flexibilité maximale.

Deux versions supplémentaires de l'A340, l'A340-500 et l'A340-600, ont été développées en collaboration étroite avec les compagnies aériennes. L'A340-600 a obtenu la certification en mai 2002 et est entré en service aérien en août alors que son frère, l'A340-500, a obtenu la certification en Décembre 2002.

En Décembre 2000, Airbus a lancé le programme A380 de 555 places à l'extrémité supérieure du spectre. Ce tout nouveau avion double pont est l'avion le plus avancé, spacieux et efficace, et la solution à la croissance du trafic aérien mondial.

L'A380 a un coût d'exploitation de 15-20% inferieur par rapport à la moyenne, une cabine de 10-15% plus large, une consommation moindre de carburant au passager-kilomètre-transporté, est moins bruyant et émet moins d'émissions que le plus gros avion en service jusqu'à aujourd'hui.

Le dernier projet d'Airbus jusqu'à aujourd'hui, l'A350 Xtra Wide Body, un avion de long range avec une capacité de 270-355 passagers, a été lancé en 2007 et la mise en service a été prévue en 3 ans 2013, 2014, 2015 pour les 3 A350-900, A350-800, A350-1000.



LA FAMILLE DES AVIONS CIVILS AIRBUS



Figure 2 - La famille des avions d'Airbus

LA FAMILLE A320 (MONO COULOIR)

La famille mono couloir contient l'A318, l'A319, l'A320, l'A321.

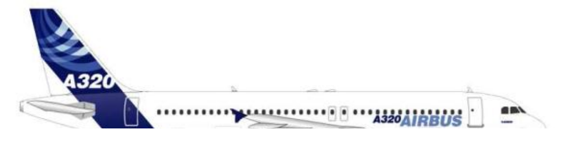


Figure 3 - L'A320

Avec plus de 6.500 vendus, plus de 4.000 en service, la famille A320 est la best-seller des avions d'Airbus. Depuis l'introduction des avantages de « fly-by-wire », l'innovation a continuellement amélioré les performances, réduit la consommation de carburant, les émissions et les niveaux de bruit à travers la famille A320.

Le plus large fuselage mono couloir du marché permet de mettre des sièges plus larges et avec plus de confort, des allées plus larges pour des temps d'escale réduits, les arrimages plus en soutes permettant l'utilisation de conteneurs de fret.

La famille A320 est un même concept d'avion dans quatre tailles, avec la même section: les A318, A319, A320 et A321 ont les mêmes systèmes, cockpit, procédures d'exploitation et de maintenance et de qualification de type pilote.

Couvrant le marché de 100 à 220 sièges et volant jusqu'à 3.700nm/6.850km, la famille A320 a un attrait indéniable pour le marché et un incomparable succès mondial.

LA FAMILLE A340 (LONG-RANGE)

La famille contient l'A330 et l'A340.



L'A330 est l'Airbus à large fuselage le plus vendu avec plus de 1.000 commandes et est le plus populaire sur le créneau des 220 à 330 sièges moyens et long-courriers aujourd'hui. Il est exploité par plus de 80 opérateurs et atterrit dans plus de 300 aéroports chaque semaine.



Figure 4 - L'A340

L'A330 polyvalent permet une rentabilité pour tous les transporteurs à faibles coûts et de charter comme pour les transporteurs des plus grands réseaux. Les passagers bénéficient de l'habitacle spacieux, calme et confortable fournis avec les services de passagers les plus avancés.

Avec près de 400 avions vendus, la famille A340 offre une vraie solution de long-courrier à plus de 45 clients et opérateurs. Les trois membres de la famille A340 peuvent transporter 300 à 400 passagers dans une cabine de grand confort sur les lignes long à très long courriers.

La famille A340 est présente sur les parcours les plus exigeants grâce à ses excellentes performances sur des aérodromes difficiles et effectue les plus longues routes exploitées aujourd'hui.

LA FAMILLE A350 WXB (WIDE EXTRA BODY)

L'A350 XWB est une nouvelle marque, l'état de l'art, de la famille des avions à long rayon d'action qui répond aux besoins de la capacité de 270 à 350 passagers.

L'A350 XWB allie des matériaux de pointe avec les systèmes les plus modernes pour assurer une efficacité inégalée et des niveaux sans précédent de confort des passagers.



Figure 5 - L'A350

La faible consommation de carburant et la réduction des coûts d'exploitation gardent à l'A350 XWB une longueur d'avance sur le 787 et d'une génération au-delà les 777.

L'impact environnemental est réduit au minimum par les émissions extrêmement faibles et les caractéristiques du bruit de l'A350 XWB, assurant ainsi un avenir durable pour l'aviation.

LA FAMILLE A380

L'A380 est l'avion le plus spacieux d'Airbus et dans le monde des avions civils. Sa section fournit les cabines flexibles et innovantes sur les deux ponts, offrant des sièges plus larges, des allées plus larges et plus d'espace au sol. La cabine ultra silencieuse, avec des sièges plus larges pour tous les passagers, la qualité de l'air rafraîchissant et un éclairage d'ambiance sensible, définissent de nouveaux standards dans le confort des passagers.



Figure 6 - L'A380



Répondant aux exigences de bruit et d'émission les plus exigeantes, compatible avec les infrastructures aéroportuaires existantes et offrant la plus faible consommation de carburant par siège dans la classe de gros porteurs, l'A380 définit de nouvelles normes économiques et environnementales.

Depuis l'entrée en service en Octobre 2007, l'A380 s'est révélé être un argument commercial vis-à-vis des passagers remarquables, en stimulant la demande de transport. C'est l'effet sur lequel les compagnies aériennes exploitant l'A380 parient le plus: un coefficient de remplissage moyen plus élevé et des profits plus élevés.

L'A380 est devenue la nouvelle icône de l'aviation, le fleuron du XXIe siècle.

A coté des avions civils, Airbus construit aussi des avions militaires (A400-M, la famille MRTT, C295, CN235, C212), des avions cargos (Beluga) et des avions privés.

L'ENVIRONNEMENT DU STAGE

AIRBUS TOULOUSE - LE SITE GUYNEMER

Le siège social d'Airbus est situé à Blagnac, une petite ville à côté de Toulouse. Ce siège se compose de plusieurs site : Blagnac (laboratoire, maquettes, unité électrique et unité composites), Saint Martin (bureau d'études, assemblage la famille A320, A400-M et livraison), Louis Breguet (logistique), Clément Ader (assemblage et équipement des long courriers (A330/A340), Lagardère (assemblage A380), Saint Eloi (assemblage des mâts moteurs), Guynemer (Support Client).

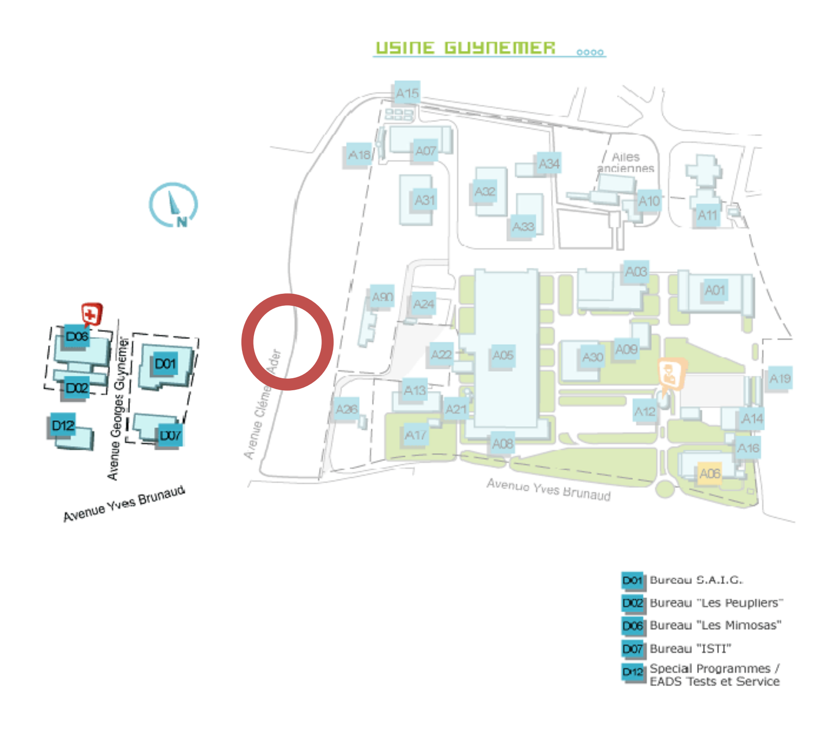


Figure 7 - Le site Guynemer

J'effectue mon stage sur le site Guynemer qui porte le nom d'un pilote de guerre français. C'est le plus petit site de Toulouse, il rassemble les activités diverses :



- Publications techniques, qui produit la documentation qui accompagne un avion tout au long de sa vie. Cette documentation explique à l'équipage le fonctionnement des avions et fournit les instructions d'entretien. Pour ce faire, une centaine de manuels sont régulièrement mis à jour et personnalisés.
- Traduction des contrats de vente et de la documentation technique.
- Les centres de formation, où divers cours de formation (langues, informatique) sont fournis aux personnels d'Airbus qui souhaitent améliorer leurs compétences ou en acquérir des nouvelles.
- Un magasin de logistique (annexe au centre Louis Breguet Athos), contenant le «hardware» des stocks.
- Le bâtiment PEGASE, le logement des bureaux BZ (programmes spéciaux).

Mon bureau est dans le bâtiment D06 (Bureau « Les Mimosas »), à la salle 392 au 3e étage, où travaillent aussi tous les employés de SEV (SEVC, SEVO, SEVS, SEVD).

SE - CUSTOMER SERVICES

Le service client (SE) est un service d'Airbus dont la mission globale est de fournir un soutien aux clients pour toutes les questions liées à l'ingénierie, de service technique et toutes les données techniques nécessaires. Le service client est composé par plusieurs sous-services :

- **Engineering Support (SEE):** Fournir un soutien technique afin d'assurer une exploitation sûre, fiable et économique de tous les avions Airbus en particulier à travers la prestation de soutien technique aux problèmes en service.
- **Upgrade Service (SEU):** Soutenir la réussite commerciale des clients d'Airbus en fournissant des mises à jour personnalisées des avions, hautement concurrentielles, des solutions de modernisation dans le cadre d'options payantes.
- **Aircraft Embodiment Operations (SEO):** Gérer et livrer à temps, au coût et au niveau de qualité appropriés aux activités industrielles, les services de réparations majeures liées à la conversion et à maintenance de l'avion.
- **Maintenance Programmes and Services (SEM):** Contribuer aux opérations des avions sûrs, fiables et économiques, en termes de programme de maintenance, d'ingénierie et de planification de l'entretien, incluant l'optimisation et l'outillage, ainsi que des publications Bulletin de service.
- **Technical Data (SED):** Fournir toutes les solutions de données techniques et de documentation pour assurer des opérations sûres, fiables et économiques de tous les avions Airbus.
- **Engineering & Maintenance Projects (SEV):** Fournir aux clients des solutions E & M (Engineering et Maintenance) cohérentes et à valeurs ajoutées pour appuyer les opérations sécuritaires, fiables et économiques de tous les avions Airbus, coordonner et surveiller tous les projets E & M pour assurer la livraison en temps opportun des résultats du projet
- **Engineering & Maintenance Quatily (SQD)**: Assurer et soutenir l'entreprise pour répondre aux différentes exigences des clients, des autorités et des normes de qualité (EN9100). En outre, promouvoir et soutenir la mise en place d'une démarche d'amélioration continue et garantir la qualité de tous les livrables et les services de l'organisation E & M, y compris ceux des sous-traitants.

SEV - ENGINEERING & MAINTENANCE PROJECTS

Le service SEV est une partie de SE (Customer Service) qui a mission de fournir des produits logiciels orientés Maintenance & Ingénierie et des services pour les opérations des avions Airbus, de coordonner la maintenance et l'ingénierie des systèmes d'information et des projets visant à fournir des résultats à valeurs ajoutées .



Les sous-services de SEV sont **SEVC** (E&M Projects L2 Support), **SEVD** (E&M Research & Data Defintion), **SEVS** (E&M Product Definition) et **SEVO** (E&M Project Platform).

Les objectifs de SEV sont de prendre en charge des produits et des services innovants grâce à des recherches ciblées ; de représenter les intérêts d'Airbus dans les groupes internationaux de normalisation des données techniques ; de définir, déployer et supporter, conduire sur le marché (market leading) des produits de Maintenance et Ingénerie (M&E) et des services aux clients et de coordonner les projets IS M&E internes en temps opportun.

SEV fournit les livrables des résultats des recherches et d'expertise liés aux données de Maintenance & Engineering, des systèmes d'information et des logiciels produits ; des spécifications de structure des données et guides de l'utilisateur liées aux données de Maintenance & Engineering ; des produits logiciels et services pour les clients de toutes unité assurant la Maintenance & Engineering des avions Airbus (les produits logiciels comme la famille AirN@v, M&E Community et des services en ligne, Repair Manager, l'ADOC Maintenance & Engineering… ou les services de déploiement, support, formation, gestion du changement, users clubs) ainsi que la consolidation des exigences, la planification intégrée, et le versioning de projets de système d'Information pour Maintenance & Engineering.

SEVS - Maintenance & Engineering Product Definition

Le service SEVS est celui dans lequel j'effectue mon stage durant 5 mois. Etant composant de SEV, il s'agit d'une équipe de 7 personnes dirigé par Patrick GUEHL, installé au 3e étage du bâtiment D06 du site de Guynemer d'Airbus. Les autres membres de cet équipe sont Jean BOUSQUET (mon maître de stage), Nathalie SIGNES, Julian BARRERA-ESQUINAS, Elisabeth DUSCH, Jean-Louis LAMOTTE, Christian GIGOT et Françoise LAMAILLOUX. Chaque membre gère certains projets en tant que chef de produit. Leurs objectifs communs sont :

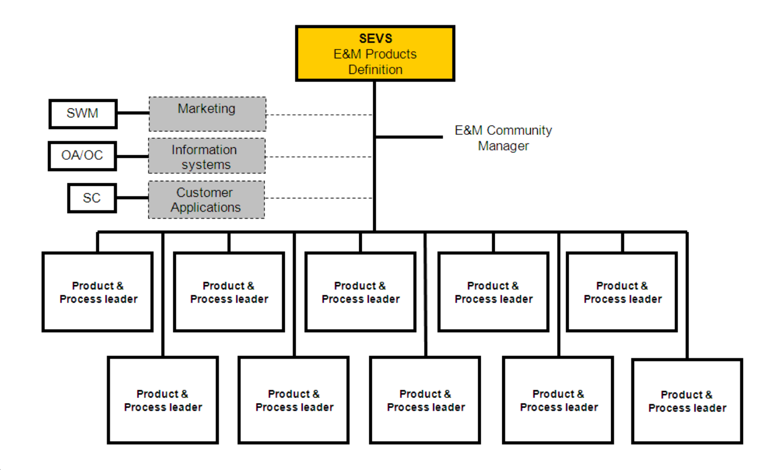


Figure 8 - Le service SEVS

- Définir et maintenir le référentiel des processus E&M des compagnies aériennes et assurer que le produit E&M correspond aux attentes des compagnies aériennes.
- Définir les produits de E&M et suivre leur développement, en particulier d'établir des spécifications techniques générales et effectuer la gestion des produits et de suivi global et le suivi des projets de développement des services.
- Coordonner toutes les activités liées au développement et l'industrialisation des lignes de production pour les services numériques de E&M.
- Gérer la communauté Maintenance & Engineering disponible dans le portail Airbus (AirbusWorld), y compris les activités telles que développer, maintenir, rédiger le contenu de la communauté.
- Proposer et participer à la définition et au suivi des projets de recherche liés.
- Proposer une stratégie détaillée des services basés sur l'ensemble des stratégies de E&M.
- Contribuer à la vision globale de E&M en déduire le catalogue de Customer Solutions.
- Établir des lignes directrices de spécification et de développement pour les services E&M et établir des méthodes et des processus pour les spécifications du e-Services.



- Assurer la cohérence inter-domaine des services de E&M.
- Fournir le support avant-vente pour des solutions E&M (famille ADOC, Airman…)
- Représenter E&M auprès des organismes de normalisation si nécessaire.

LES PRODUITS M&E

Les produits de Maintenance&Engineering gérés par SEVS sont : AJCP(*), Electronic Job Card, Engineering Drawing, Mechanical Drawing, LATC(*), LETD(*), M&E Community, OLCM(*), PACAT(*) ADOC, PACAT AirN@v, TDSF(*) Merge Tool, la famille ADOC et la famille AirN@v... La suite de l'exposé ne s'intéresse qu'aux deux familles de produits : la famille ADOC et la famille AiN@v.

LA FAMILLE ADOC



ADOC est une solution globale et modulaire d'Airbus apportée aux clients pour la gestion de documentation numérique (les Job Cards, les pages de résultat de recherche…). ADOC permet d'effectuer efficacement la gestion de documentation, d'économiser le temps et les coûts dans les domaines de Maintenance & engineering et Flight Operations. Les solutions ADOC peuvent être implémentées comme un projet global ou étape par étape, d'une structure simple pour la consultation des documentations numériques ou la production des Job Cards à un système d'information de maintenance entièrement intégré. ADOC est déjà utilisé par 260 compagnies aériennes y compris Air France, Virgin Atlantic, Fly Emirates, jetBlue Airways, Cathay Pacific, Air Canada…

La famille ADOC se compose des modules : **ADOC Manager** (pour la gestion des documents de n'importe quel type d'avion, moteur ou les documents internes), **ADOC Revision Manager** (pour le transfert les modifications internes des documents à la prochaine révision venant du constructeur), **ADOC N@vigator** (pour la consultation avancée des données techniques d'Airbus et non-Airbus, basé sur la base de donnée de la consultation d'ADOC Electronic Publisher), **ADOC Job Card Publisher** (pour la production de Job Cards créées par le système d'information de Maintenance) et **ADOC Electronic Publisher** (pour la préparation de base de données de consultation et la personnalisation d'application de consultation avancée).

Les 3 solutions principales intégrées proposées par la famille ADOC sont :

- ADOC Content Management : pour améliorer la mise à jour, le processus et la personnalisation des manuels (créer, mettre à jour les documents personnels). Les deux modules sont comportés : ADOC Manager et ADOC Revision Manager.
- ADOC Consultation : augmenter la vitesse et l'efficacité de recherche, économiser le temps de recherche. La consultation est faite en naviguant sur les documentations en SGML et XML à l'aide de ADOC N@vigator.
- ADOC Job Card Publisher : simplifier la production des Job Cards (automatisation de la production des Job Cards, personnalisation des Job Cards avec les outils intégrés). Cette solution utilise le module Job Card Publisher.



LA FAMILLE AIRN@V

Basé sur la technologie de ADOC N@vigator, AirN@v est une solution standard pour la consultation avancée des données techniques de Maintenance & Engineering Airbus. Cela permet la récupération facile et précise des informations, une navigation et des fonctions de recherche efficaces en utilisant la dernière technologie de web. AirN@v est publié avec les outils de la famille ADOC, livré en DVD et aussi accessible en ligne par AirbusWorld. La famille AirN@v est composée de AirN@v Maintenance, AirN@v Repair, AirN@v Workshop, AirN@v Planning (les documents sur le planning des maintenance), AirN@v Engineering (les documents d'ingénerie) et AirN@v Associated Data (les documents sur la liste des consommable, des outils et équipements standards).



AirN@v Maintenance est disponible pour tous les programmes d'Airbus sauf A300 B2/B4, cela permet aux mécaniciens et aux ingénieurs de consulter et de naviguer parmi les documents principaux de Maintenance comme le dépannage, les catalogues des parts illustrés, le câblage de l'avion…), en incluant les analyses des pannes, les images interactives, les vidéos…

AirN@v Repair est disponible pour les programmes de A320, A330, A340 et A380, cela permet les mécaniciens et aux ingénieurs de consulter et de naviguer parmi les documents de réparation et dépannage structurels avec les manuels SRM (Structural Repair Manual) et NTM (Non destructive Testing Manual).

AirN@v Workshop est disponible pour tous les programmes d'Airbus, permet la consultation et la navigation parmi les documents d'atelier contenant les manuels de maintenance des composants et la réparation des conduits et des tuyaux de carburant avec des manuels CMM (Component Maintenance Manual) et DPFRM (Duct and Fuel Pipe Repair Manual).

AirN@v Planning est disponible pour tous les programmes d'Airbus sauf A300 B2/B4, cela permet de naviguer et de consulter les documents sur le planning de maintenance.

AirN@v Engineering permet de consulter et naviguer parmi les documents d'ingénérie comme les Services Bulletins…

AirN@v Associated Data est disponible pour tous les programmes d'Airbus cela permet de naviguer et de consulter les documents sur la liste des consommables, des outils et équipements standards.



SYNTHÈSE DES TRAVAUX RÉALISÉS

ANALYSE PRÉALABLE DU PROJET

Cette étude, décomposée en cinq sous-étapes, consiste à déterminer l'objectif principal du projet puis analyser les applications existantes afin d'en retirer les différents inconvénients. Par la suite sera exposé, le cahier des charges qui présente les exigences de l'entreprise face à la future application.

INTRODUCTION DU PROJET

SUJET DE STAGE

Le sujet du stage est de réaliser une étude de convergence entre la phase de recueil des exigences et la phase de validation d'un projet, en vue d'harmoniser le processus, les fichiers utilisés et de proposer une solution pour réutiliser les données entre les deux outils utilisés pour chacune des phases.

Plus précisément, mes missions sont d'étudier sur la traçabilité entre les exigences du BRD (Business Requirement Dossier) et les Use Case de Validation d'un projet et de développer un outil qui génère de façon automatique ces liens de traçabilité. Ces missions sont réalisées en divisant en plusieurs étapes de différents types (analyse des besoins, étude des règles, développement, test et évolution) que j'exposerai dans les prochaines parties.

MATÉRIEL

Lors de le premier jour de mon arrivée dans l'entreprise, j'ai été accueilli par les ressources humaines afin de signer les documents relatifs à la sécurité et la confidentialité et obtenir un badge donnant accès aux différents sites et services au sein d'Airbus. J'ai ensuite été conduit dans le bâtiment du service SEVS (bâtiment D06 du site Guynemer) pour définir avec mon maître de stage M BOUSQUET Jean et mes collègues du service SEVS les taches que j'aurais à accomplir et m'installer dans le bureau qu'on m'avait alloué.

Dans le cadre de mon stage, je dispose d'un bureau qui se situe au 3e étage du bâtiment D06, dans la même salle de tous les membres du service. Mon poste a été installé avec le système d'exploitation de Windows XP Professionnel SP3 version 2002, avec les outils bureautiques de base comme Microsoft Office 2007, Outlook 2007 et les logiciels professionnels comme Borland CaliberRM, ISO View 5. La connexion d'Internet est aussi disponible sur mon poste car chaque jour je dois travailler avec les outils à distance via Internet Explorer (ce sont les outils installés dans le serveur de l'intranet, pas directement dans mon poste) comme HPQC et les outils d'Airn@v (Maintenance, Associated Data, Workshop…).

Les deux outils qui sont les plus importants et qui ont d'impact direct avec mon sujet de stage sont Borland CaliberRM et HPQC, j'en parlerai plus précisément dans la partie « *Applications existantes* » p.26. En plus, il existe une liste des autres logiciels (Oracle, Eclipse…) prêts à installer sur ServiceDesk.



CONTEXTE

LA GESTION DES PROJETS INFORMATIQUE

Selon wikipédia, la **gestion de projet** (ou *conduite de projet*) est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Lorsque la gestion de projet porte sur un ensemble de projets concourant à un même objectif, on parle de gestion de programme.

Un projet en général est un ensemble finalisé d'activités et d'actions entreprises dans le but de répondre à un besoin défini dans des délais fixés et dans la limite de l'enveloppe budgétaire allouée. Un projet informatique est un projet appliqué dans le domaine informatique, ce qui revient souvent à développer un outil, une fonctionnalité d'une application… Les projets d'Airbus gérés par SEVS sont les projets autours la famille AirN@v (Maintenance, Associated Data, Workshop, Repair, Planning, Engineering) et aussi les projets particuliers (LETD, AirPl@n Simulator, LETD, LATC, TDSF Merge Tool, Electronic Job-Card…). La méthode de gestion de projet utilisée est le cycle de vie en V.

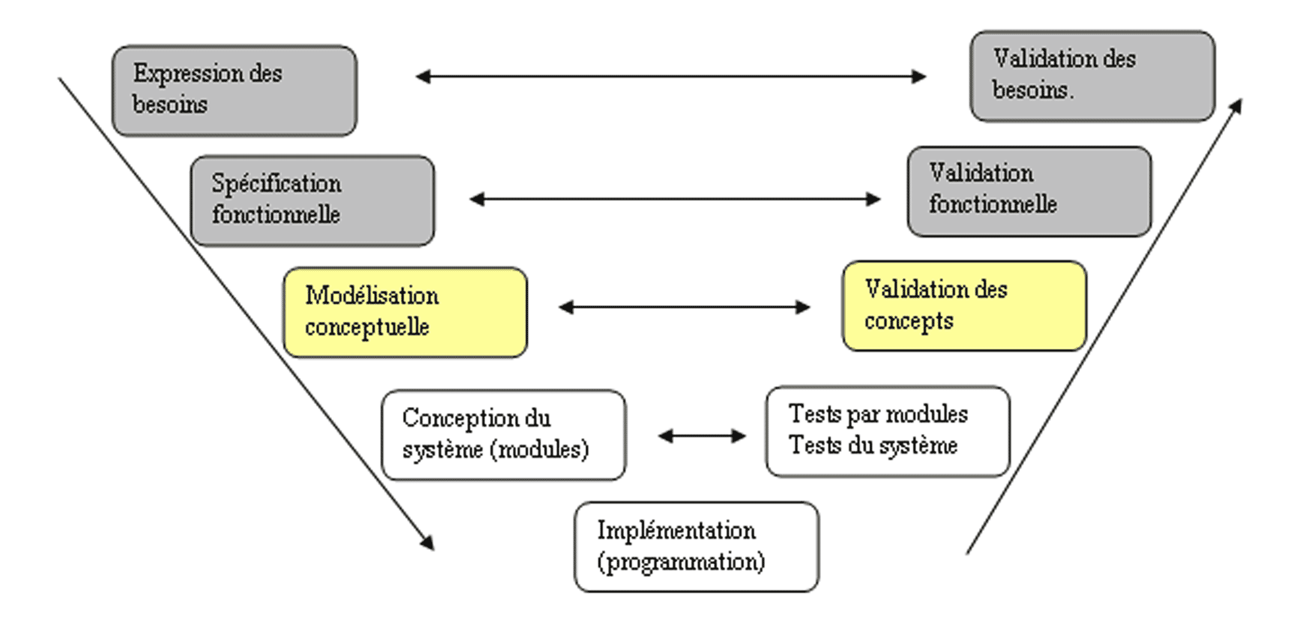


Figure 9 - Le cycle de vie en V

Le modèle de cycle de vie en V part du principe que les procédures de vérification de la conformité du logiciel aux spécifications doivent être élaborées dès les phases de conception.

La représentation en V tient d'avantage compte de la réalité, le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles.

Elle montre que:

- c'est en phase de spécification que l'on se préoccupe des procédures de qualification.
- c'est en phase de conception globale que l'on se préoccupe des procédures d'intégration.
- c'est en phase de conception détaillée que l'on prépare les tests unitaires.



Le modèle de cycle de vie en V permet d'anticiper sur les phases ultérieures de développement du produit. En particulier le modèle en V permet de commencer plus tôt:

- Plan de tests de qualification.
- Plan d'évaluation des performances.

Dans le modèle en V, le chef de produit (le MOA) prend en charge la première étape et la dernière étape : la phase de recueil des exigences et la phase de validation.

V&V METHOD

PRINCIPE

Dans la gestion de projets de logiciels, le test de logiciels, et le génie logiciel, la vérification et la validation (V&V) est le processus de vérification qu'un système logiciel répond aux spécifications et qu'il remplit son objectif. Il fait normalement partie du processus de tests de logiciels d'un projet.

La méthodologie V&V (V&V methodology) est une partie du processus de tests de logiciels d'un projet. Elle se compose en 2 parties : la vérification et la validation.

La validation est le fait de confirmer, à travers l'évidence des objectifs, que les exigences d'un Intended Uses (prévu) ou d'une application ont été bien remplies. Cette étape va répondre à la question « Est-ce qu'on a construit le bon produit ? », c'est-à-dire le logiciel doit faire ce dont l'utilisateur a besoin.

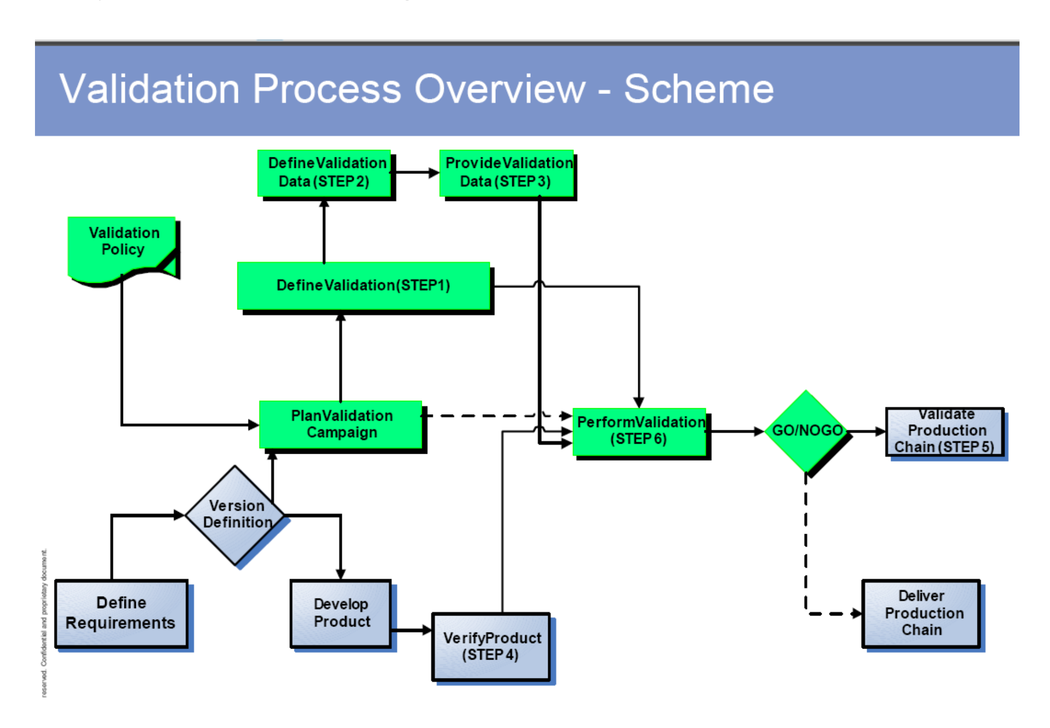


Figure 10 - Schéma du processus de validation

La vérification est le fait de confirmer, à travers l'évidence des objectifs, que les exigences spécifiées ont été bien remplies. Cette étape va répondre à la question « Est-ce qu'on a bien construit le produit? », c'est-à-dire le logiciel doit être conforme à sa spécification.



LES DÉFINITIONS

Les acteurs du processus:

Acteurs	Rôle
Business Representative	Exprimer les Intended Uses
Product Leader (Chef de produit)	Définir la politique de la Validation, rédiger les Use Cases et Test Cases et est responsable de la décision de Go/No Go finale
Data Designer	Spécifier les données utilisées dans la validation du produit
Data Provider	Fournir les données de validation
Validation Manager	Planifier et Gérer la campagne de validation
Validation Performer	Lancer les Tests et fournir les résultats de Tests

Une campagne de Validation est une liste d'activités permettant de valider une certaine version d'un produit. Une décision de Go/No Go sera prise à la fin d'une campagne de validation. La campagne de Test peut identifier des problèmes : les problèmes importants pourraient conduire à une décision de No Go, ce qui doit être immédiatement communiquer au chef de produit, les problèmes mineurs pourraient être empilés jusqu'à la fin de la campagne.

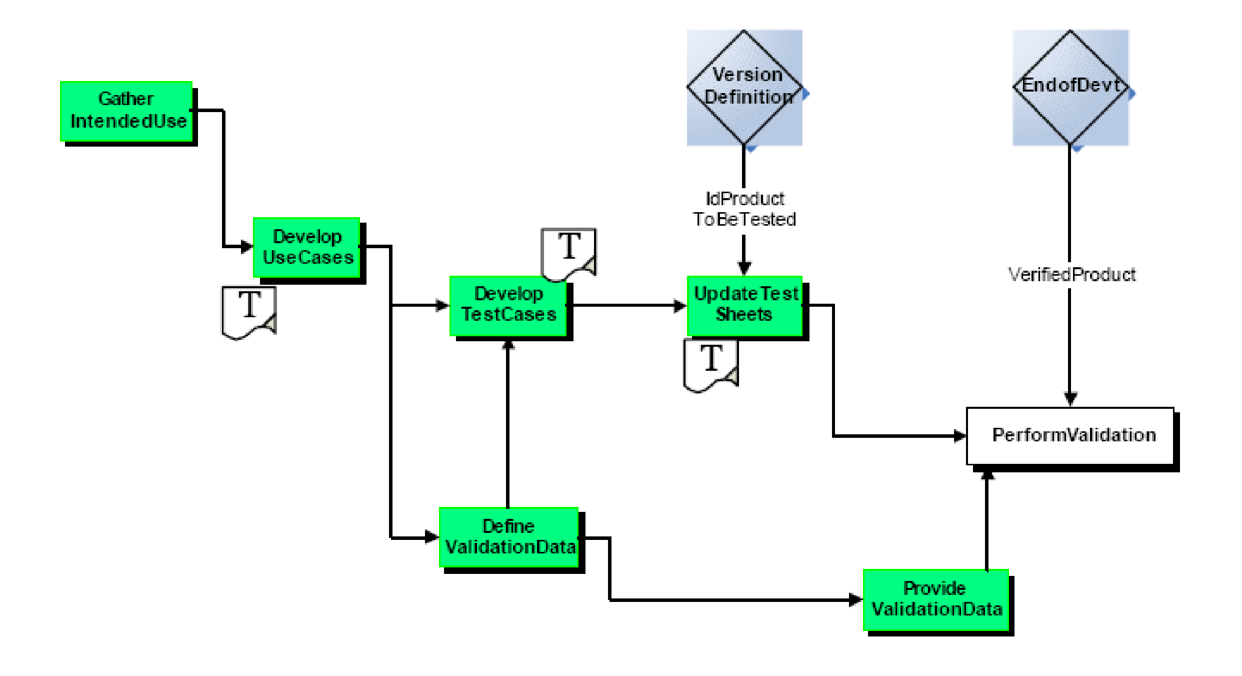


Figure 11 - Campagne de validation

Les éléments (outils) utilisés dans une campagne de Test sont :

<u>Intended Use Case</u> : décrit les choses telles que l'application d'un modèle objet, le contexte ou partie du cycle de développement dans lequel le modèle objet est destiné à être utilisé, le niveau d'abstraction des objets définis à l'aide du modèle, et le but de l'orientation d'objet dans l'application prévue. Les Intended Uses sont fournis par le client de l'application.



Par exemple, quelques Intended Uses du projet Workshop V3 sont « Search Item by P/N », « Internal Links », « TOC & View »…

Use Case : est une description des étapes ou des actions entre un utilisateur (ou «acteur») et un système logiciel qui conduit l'utilisateur vers un résultat attendu du système. Un Use Case est une formalisation d'un chemin d'un Intended Use selon le scénario de réalisation. Ils sont formalisés sous la responsabilité du chef de produit. Un Intended Use peut avoir plusieurs Use Cases, par contre, dans certains cas, on peut avoir un Intended Use et un Use Case commun.

Par exemple : pour l'Intended Use « Search Item by P/N » du projet Workshop, on a un Use Case commun nommé « Search Item by P/N », par contre, pour l'IU « Internal Links », on peut avoir plusieurs Use Cases qui définissent plusieurs liens internes inclus dans l'application, ou les Use Cases de l'Intended Use « TOC & View » représentent les différents éléments de la table des matières comme Page Block, Task, Topic, Highlight, Detailed Part List Display…

Test Case : est un ensemble de conditions ou variables pour lesquelles un testeur va déterminer si une application ou un système logiciel fonctionne correctement ou non. Un Test Case est un chemin d'un Use Case contenant les données de Test (input ou les paramètres de contrôle).

LA TRAÇABILITÉ

Les Use Cases décrivent les exigences fonctionnelles et des spécifications supplémentaires décrivent les éléments non-fonctionnels. En outre, tous les cas utilisent des maps pour de nombreux scénarios. Le mapping des Use Cases à des scénarios est donc une relation un à plusieurs. De la même façon, le mapping des scénarios aux Test Case est aussi relation un à plusieurs. Entre les besoins et les exigences, d'autre part, il y a une relation plusieurs à plusieurs.

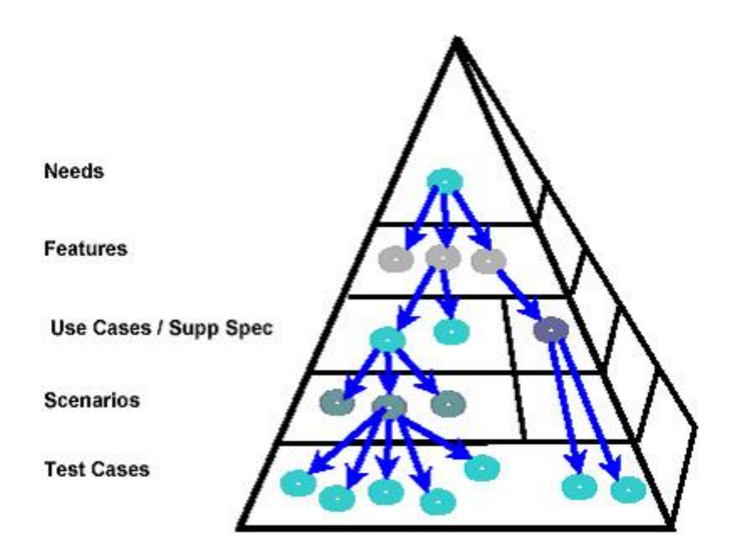


Figure 12 - Pyramide de traçabilité

La traçabilité joue plusieurs rôles importants:

- Vérifier que l'implémentation remplit toutes les exigences: Tout ce que le client demande a été appliqué.
- Vérifier que l'application ne fait que ce qui était demandé: Ne pas appliquer ce que le client n'a jamais demandé.



- Aider à la gestion du changement: Lorsque certaines exigences changent, nous voulons savoir quels cas de test doivent être refait pour tester ce changement.

APPLICATIONS EXISTANTS

BORLAND CALIBERRM

Le progiciel CaliberRM est utilisé par SEVS pour la gestion des exigences des projets.

Borland® CaliberRM™ est une plate-forme de gestion des exigences logicielles facilitant la collaboration, l’analyse d’impact et la communication afin de sécuriser la précision et la prévisibilité des livraisons clés du projet. CaliberRM aide également les organisations petites, grandes ou distribuées à s’assurer que les applications répondent aux besoins des utilisateurs finaux en permettant aux analystes, développeurs, testeurs et autres intervenants de capturer et communiquer l’avis des utilisateurs tout au long du cycle de vie des applications.

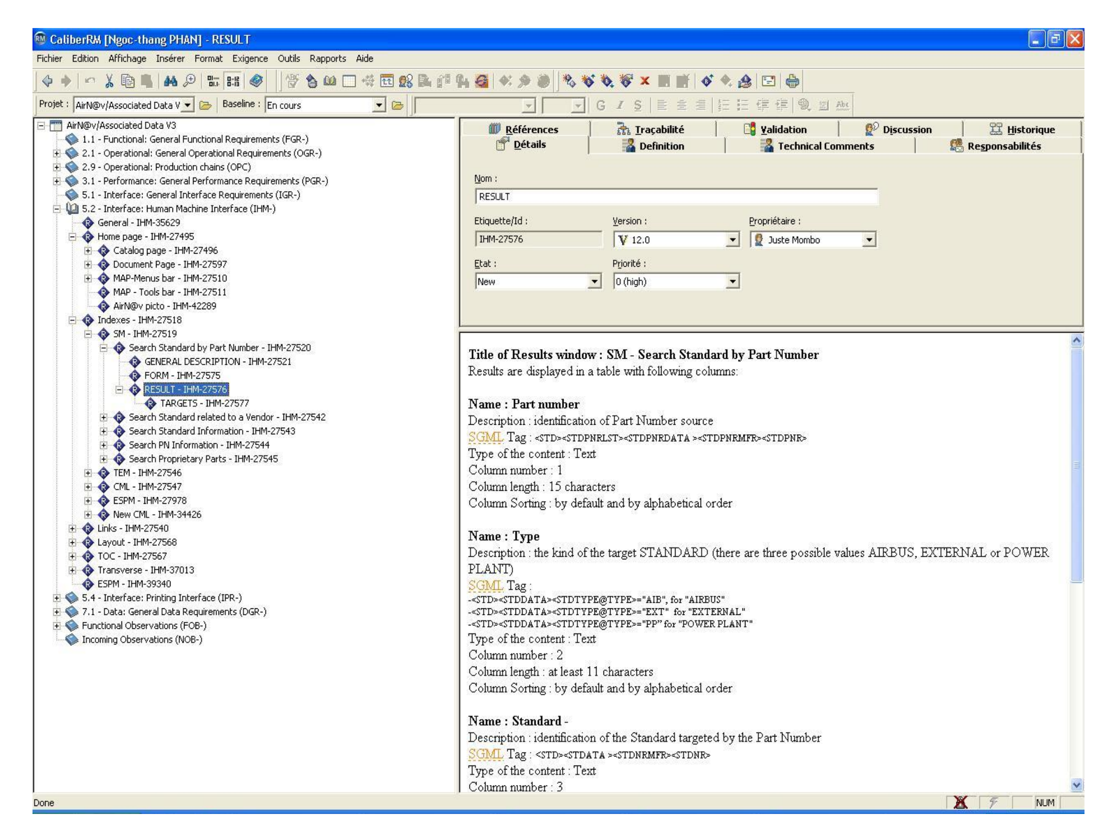


Figure 13 -L'interface de CaliberRM

Les fonctionnalités avancées de gestion des exigences sont :

- **Référentiel central**: Borland CaliberRM propose un référentiel central sécurisé pour gérer l’ensemble des exigences projet.

- **Adaptabilité**: Borland CaliberRM est conçu pour s’adapter aux processus de gestion des exigences existants.

- **Traçabilité de bout en bout** : Grâce à son architecture ouverte, Borland CaliberRM permet de relier les exigences logicielles à une grande variété d’artefacts tout au long du cycle de développement.



- **Analyse globale d'impact**: De multiples modes de visualisation de traçabilité permettent aux utilisateurs d'appréhender instantanément le périmètre d'analyse requis pour juger des effets d'un changement spécifique.

- **Multiplicité des clients**: Borland CaliberRM intègre des clients pour différents utilisateurs : Web, Eclipse, Microsoft Visual Studio (y compris Team System) et Windows.

- **Intégration**: Grâce à l'intégration transparente de CaliberRM aux autres produits Borland, les exigences logicielles peuvent être gérées et contrôlées tout au long du cycle de développement.

DocumentFactory :

Le générateur de documents, Document Factory, est une fonction inclus dans CaliberRM qui permet d'utiliser des modèles Microsoft Word pour générer n'importe quel type de document de spécifications personnalisées tenues à jour.

Pour définir les exigences sur lesquelles générer le rapport, on peut :

- sélectionner une exigence dans l'arborescence.
- sélectionner des exigences dans la grille d'exigences
- spécifier des filtres dans le modèle Document Factory

Pour l'utiliser, tout d'abord, dans Word, on doit créer des modèles (des « templates » de type dot) contenant des champs qui correspondent à des attributs d'exigences. On peut ensuite utiliser Document Factory pour remplir le modèle avec des données de n'importe quel projet. A l'aide de critères de filtrage dans les champs de modèle, il est également possible de choisir quelles données d'exigence inclure dans le document. Document Factory génère des fichiers .doc, .txt et .rtf.

Dans la réalisation de l'objectif du projet, l'outil utilise DocumentFactory comme la première étape pour exporter les exigences de CaliberRM pour les traitements suivants. Le détail du fonctionnement sera exposé dans la partie *« Export du BRD avec Document Factory »* p.36.

HP Quality Center (HPQC) 9.2

HPQC est un outil web de gestion des Tests, développé par HP Software Division de Helwett-Packard. Chez Airbus, on peut accéder à Quality Center sur l'Intranet d'entreprise, via un navigateur Web (Internet Explorer).

SEVS utilise HPQC pour maintenir une base de données des projets des tests qui peut fournir les étapes de la campagne de Test comme l'analyse des Use Cases, la génération des Test Cases, l'organisation les Tests dans les groupes particuliers, la planification, l'exécution des jeux de Test, la collecte des résultats de test et l'analyse des données.

Dans la réalisation de l'objectif du projet de stage, HPQC est utilisé à la fin du processus de l'outil pour exporter les Use Cases existant pour la comparaison avec les Use Cases générés par l'outil à partir des exigences dans CaliberRM.



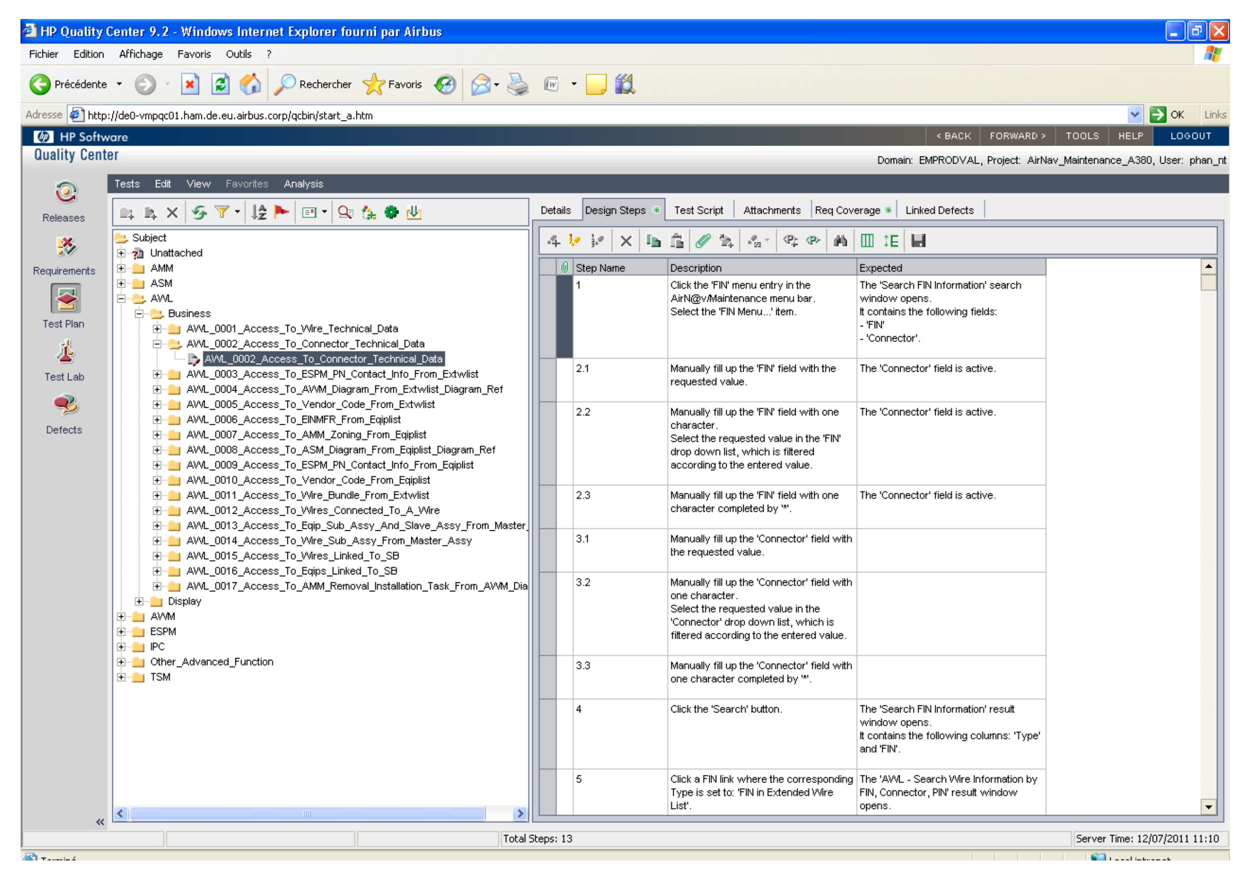


Figure 14 - L'interface de HPQC

LIMITE DES OUTILS ACTUELS

D'après l'étude de l'existant, les limites du système de gestion des exigences et des Use Cases de Test des projets sont détectées, elles sont dues surtout à la dissociation entre les 2 progiciels indépendants CaliberRM et HPQC, ainsi qu'entre les différents projets gérés par les différents méthodes (structure d'arborescence des exigences, différents fonctionnements des différents projets...

Puisque HPQC à commencé à être utiliser par SEVS beaucoup plus tard que CaliberRM (HPQC est utilisé depuis 2009 tandis que CaliberRM est utilisé depuis 2004/2005 chez SEVS), lors de la mise en place de HPQC, les exigences existaient dans CaliberRM. Donc, des exigences devaient être transférées depuis CaliberRM vers HPQC. Cette action est faite en façon manuelle, c'est pourquoi il est nécessaire de vérifier qu'on ne perd rien dans ce transfert. La couverture des exigences doit être assurée.

La structure des exigences des projets n'est pas commune. Les exigences des différents projets gérés par différents membres de SEVS sont classées en façons différentes. Ce problème peut créer les difficultés dans l'automatisation du système de gestion des exigences et des Use Cases car il n'y pas de règle commune à appliquer.

Maintenant que les outils existants ont été étudiés et que leurs limites d'utilisation ont été présentées, il est important de définir les besoins du service informatique quant à la future application.



L'OBJECTIF DU PROJET - CRITÈRES EXPRIMÉES POUR L'APPLICATION FUTURE,

L'OBJECTIF DU PROJET

La gestion des exigences, les Uses Cases et les Test Cases dans CaliberRM et HPQC est représentée par le schéma ci-dessous :

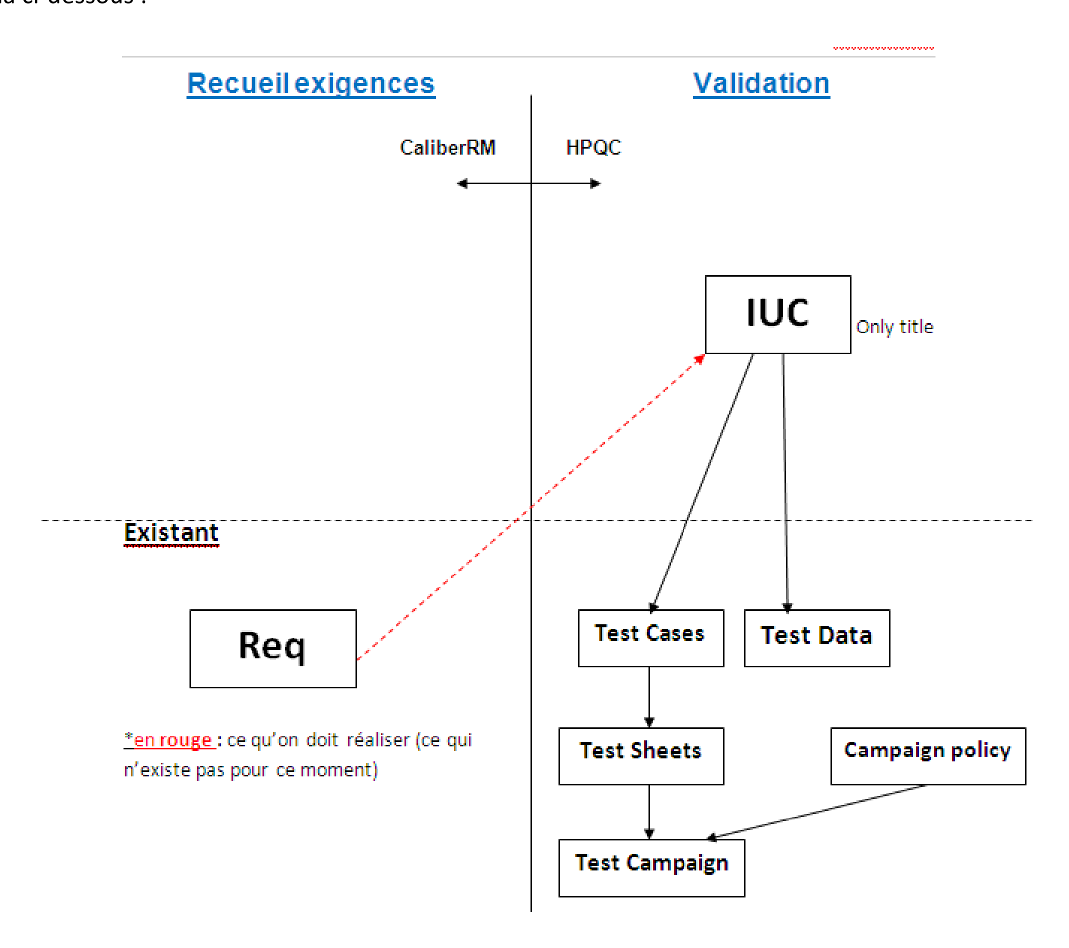


Figure 15 - Schéma de traçabilité entre CaliberRM et HPQC avant la spécification

Les exigences (Req) recueillies des clients sont incluses dans un Dossier intitulé BRD (Business Requirement Document). Ce document est rédigé dans l'outil CaliberRM pour gérer et compléter les informations nécessaires afin de livrer aux autres services (SBA ou sous-traitances) pour les étapes qui suivent (développement, test…).

A partir des exigences de CaliberRM, le chef de produit va écrire les IUC (Intented Use Cases) pour fournir les Test Cases à la Campagne de Test. Le testeur va prendre en compte des cas de Test (Test Cases) et des données de Test (Test Data) fournies par un autre service pour effectuer les tests. Les résultats des Test (y compris des bugs (Defects) sont analysés par le chef de produit afin de donner une décision Go/No Go de cette étape. Toutes ces étapes de Test sont gérées sur HPQC.

L'objectif du projet est d'assurer que la totalité des exigences de CaliberRM passe en IUC des HPQC, autrement dit, toutes les exigences sur CaliberRM doivent être couverte par au moins un Use Cases correspondants sur HPQC. Cette action était faite par le chef de produit en façon manuelle. Pour les gros projets qui ont plus que 1500 exigences, cette vérification est très compliquée et prend beaucoup de temps. Donc l'automatisation de ce processus par un outil est nécessaire.



SOLUTION GÉNÉRALE

L'étude de convergence entre la phase de recueil des exigences (BRD) et la phase de validation (IUC) s'est commencée par la nécessité de trouver une méthode pour construire automatiquement la traçabilité entre les exigences et les Intended Use Cases. La problématique principale à gérer est l'incompatibilité entre les exigences qui sont sur CaliberRM et les Intended Use Cases sur HPQC.

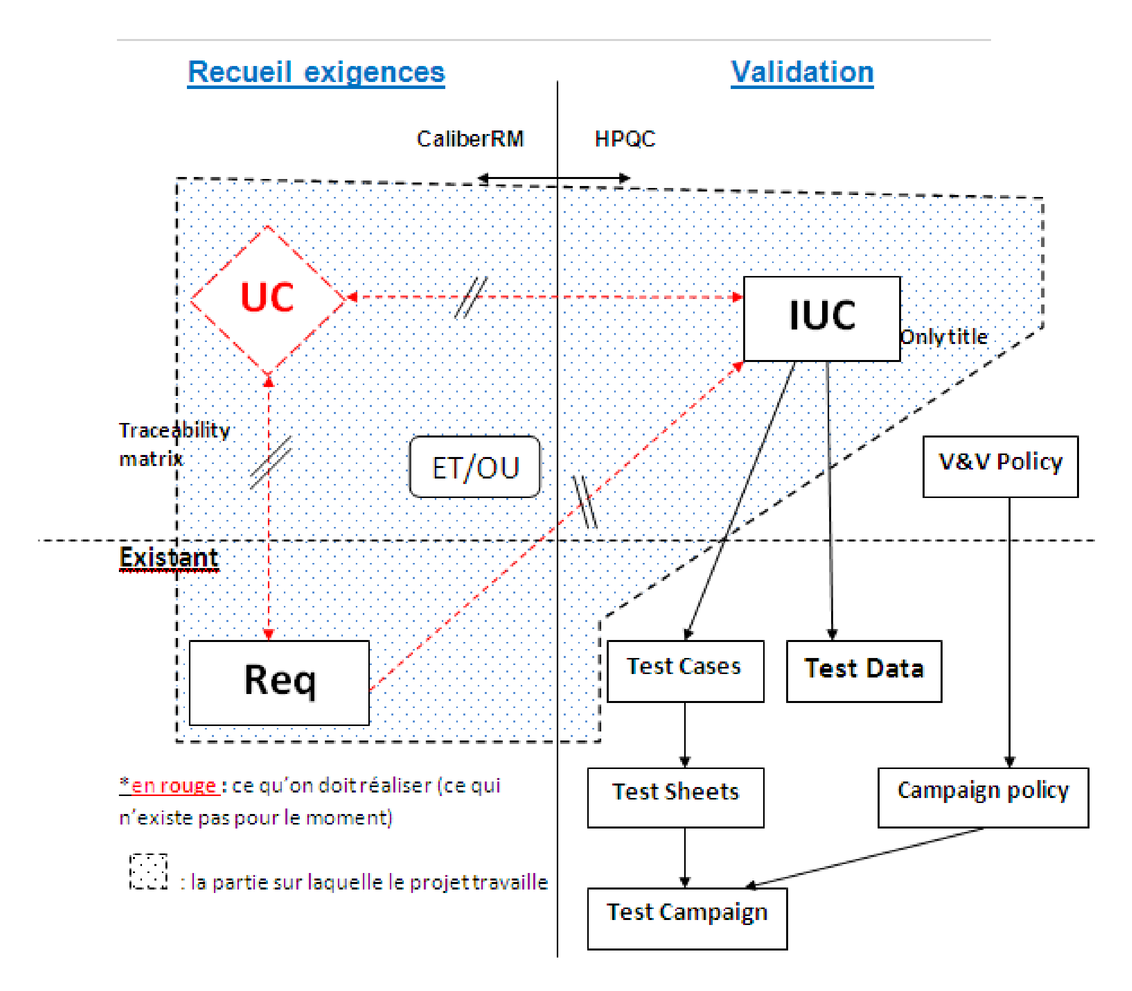


Figure 16 - Schéma de traçabilité entre CaliberRM et HPQC après la spécification

Afin de faciliter la vérification de la couverture des IUC sur les exigences, on doit générer aussi les Use Cases. Ces Use Cases jouent un rôle intermédiaire pour la traçabilité entre les exigences et les IUC (on peut voir sur le schéma).

Ces Use Cases ne sont pas livrés aux autres services, ils sont gardés par SEVS pour effectuer la vérification. Cette vérification est réalisée en construisant les matrices de traçabilité entre les différents éléments afin de calculer exactement le ratio de la couverture des exigences par les Use Cases.

La génération du nom des Use Cases sur CaliberRM doit être automatique. Pour réaliser cette automatisation, il va falloir proposer les règles qui s'appliquent sur la structure des exigences dans le BRD, leurs informations et leurs attributs gérés par le chef de produit et le logiciel CaliberRM. L'étude de ces règles a été effectuée en analysant plusieurs BRD des projets existants de SEVS.

Une fois la matrice de traçabilité construite, le calcul du ratio de la couverture sera effectué en basant sur le nombre de Use Cases générés et le nombre d'exigences du BRD. Le résultat final de ce calcul doit être présenté en exportant un fichier log. En général, ce ratio représente la performance de la génération des Use Cases (des



règles proposées sur les exigences). Le but du projet est aussi d'améliorer ces règles afin d'augmenter le ratio au maximum possible.

Dès que la liste des Use Cases est construite (après la génération et le test de couverture), elle sera utilisée pour compléter la partie « Requirement Area » dans HPQC. Pour les projets ayant déjà des IUC, l'exportation de ces IUC sera faite pour la comparaison. Pour les projets n'ayant pas encore de IUC, l'importation des UC sera effectuée. Ces Use Cases n'ont que le titre (only title) et sont stockés dans HPQC afin de servir dans la génération des Test Cases.

Pour les exigences ne pouvant pas être couvertes par les Use Cases créés automatiquement, on doit les gérer en façon manuelle.

La réalisation de cette spécification est divisée en plusieurs étapes et sera détaillée dans les parties suivantes.

EXIGENCES FONCTIONNELLES DE L'OUTIL FUTUR

Du côté analyse, il est aussi nécessaire de proposer les règles communes pour la construction des exigences dans chaque projet pour avoir une forme commune des structure de tous les projets afin de faciliter les traitements qui pourrait être effectués sur ces projets dans l'avenir. Une fois les règles établies, les nouveaux projets vont les appliquer et les anciens projets vont être remis en forme conformément à ces règles.

Du côté technique, l'outil futur doit permettre de :

- Réaliser la traçabilité entre les exigences et les Intended Use Cases de Test.
- Automatiser la génération du nom des Use Cases à partir des exigences de CaliberRM.
- Calculer la couverture des Use Cases de Test sur l'ensemble des exigences livrées.
- Rédiger un log qui joue le rôle d'un rapport résumant les résultats du processus.

Du côté performance, l'outil doit être facile à installer et à utiliser. Les étapes de lancement ne doivent pas être trop compliquées pour les non-informaticiens, surtout il faut diminuer autant que possible le nombre des étapes (des cliques et des saisis). Un guide d'utilisateur doit être rédigé en anglais, simple et compréhensible. En plus, le temps d'exécution de l'outil ne doit pas être trop long, et le taux d'utilisation du mémoire et du processus de l'outil ne doit pas être trop important afin de ne pas empêcher les autres tâches lancées simultanément dans l'ordinateur.

Tous ces critères ne sont pas considérés comme points bloquants mais ils seront abordés dans l'amélioration de l'outil dans les versions suivantes (avec l'ajout de quelques fonctions plus spécialisées demandé par les utilisateurs de l'outil).

ETUDE DES DIFFÉRENTS SCÉNARIOS

Il existe différentes façons pour réaliser l'objectif du projet, en utilisant les différents logiciels pour coder avec plus ou moins de fonctionnalités et performances. Les critères posées sont la facilité de l'utilisation et de l'installation, la performance et le temps de développement.

Deux scénarios ont été analysés. Chacun a été testé sur les différents projets pour avoir une vue sur les enjeux, les contextes de chaque solutions.



SCÉNARIO 1 : GÉRER DIRECTEMENT LES USE CASES SUR CALIBERRM

La génération des Use Cases est faite manuellement par le chef de produit en construisant le BRD dans CaliberRM. Ces Use Cases sont créés directement dans CaliberRM, sous la forme d'une exigence normale et sont souvent placés sous le type General Functional Requirement. Ces Use Cases ne sont pas livrés aux autres services, ils ne sont utilisés que dans SEVS.

Chaque Use Case a un code défini par son propriétaire (son créateur). Son contenu est aussi écrit dans CaliberRM (avec des parties Business Needs, Rationale, Pre-condition, Post-condition) comme un Use Case normal, par contre il a aussi les attributs d'une exigence normale (identifiant, structure hiérarchique…).

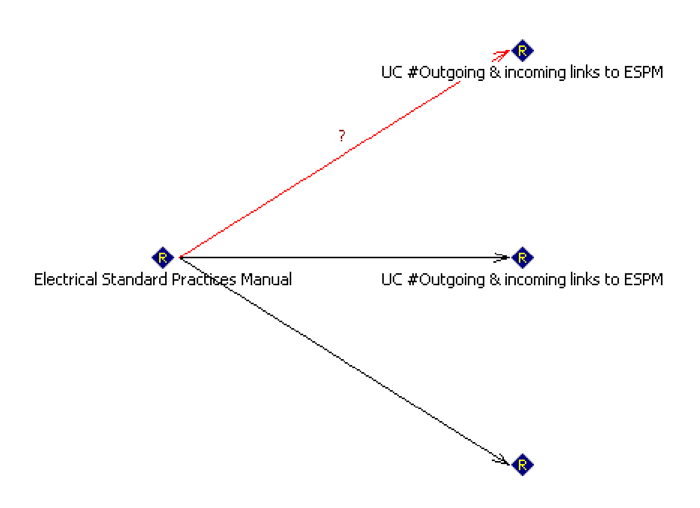


Figure 17 - Diagramme de traçabilité d'une exigence particulière

La traçabilité est aussi faite dans le sein de CaliberRM en utilisant l'onglet « Traçabilité » d'une exigence. Dans ce tab, on peut ajouter, modifier et supprimer un lien de traçabilité de l'exigence courante vers ou depuis une autre exigence. En créant le lien de traçabilité depuis un Use Cases vers ses exigences, on peut associer toutes les exigences à leurs Use Cases.

La matrice de traçabilité est un fichier .tmv qui sera exporté depuis CaliberRM, qui contient, grâce à des fonctions de filtre, tous les Use Cases qu'on a créé et les exigences associées afin d'effectuer un test de vérification de la couverture des Use Cases.

L'avantage de ce scénario est que les Use Cases sont créés directement dans CaliberRM par le chef de produit, donc il peut gérer facilement tous ces Use Cases et leurs liens de traçabilité. En plus, ce fait va réduire les erreurs venant du cas où plusieurs personnes travaillent sur les mêmes objets. La matrice de traçabilité est construite par CaliberRM, cela va réduire le temps de développement du processus.

L'inconvénient de ce scénario est aussi celui de son avantage. Puisque toutes les étapes sont faites de façon manuelle et personnelle par le chef de produit, cette personne va être surchargée des tâches, surtout pour les gros projets comme AirN@v A350 avec plus de 1600 exigences (on va voir l'analyse détaillé dans la partie d'analyse du projet A350 dans *Conception/Dossier d'analyse des projets, p.34*). En plus, la relation du projet avec les autres personnes est très limitée. Si le projet doit être passé à un autre chef de produit, cette personne aura du mal à le suivre car ses méthodes de gestion sont différentes de celles de l'ancien propriétaire du projet.



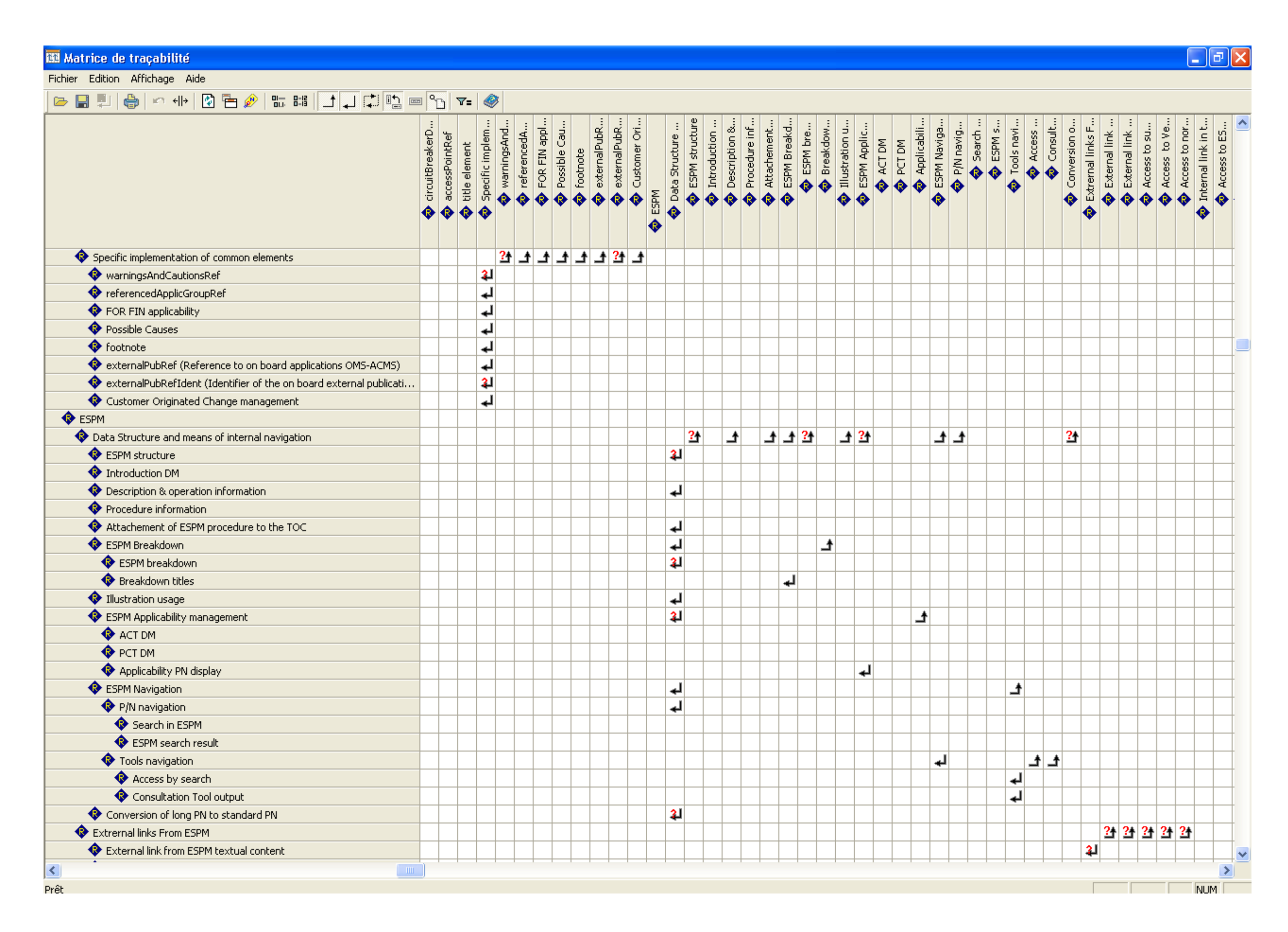


Figure 18 - La matrice de traçabilité du projet A350 fournie par CaliberRM

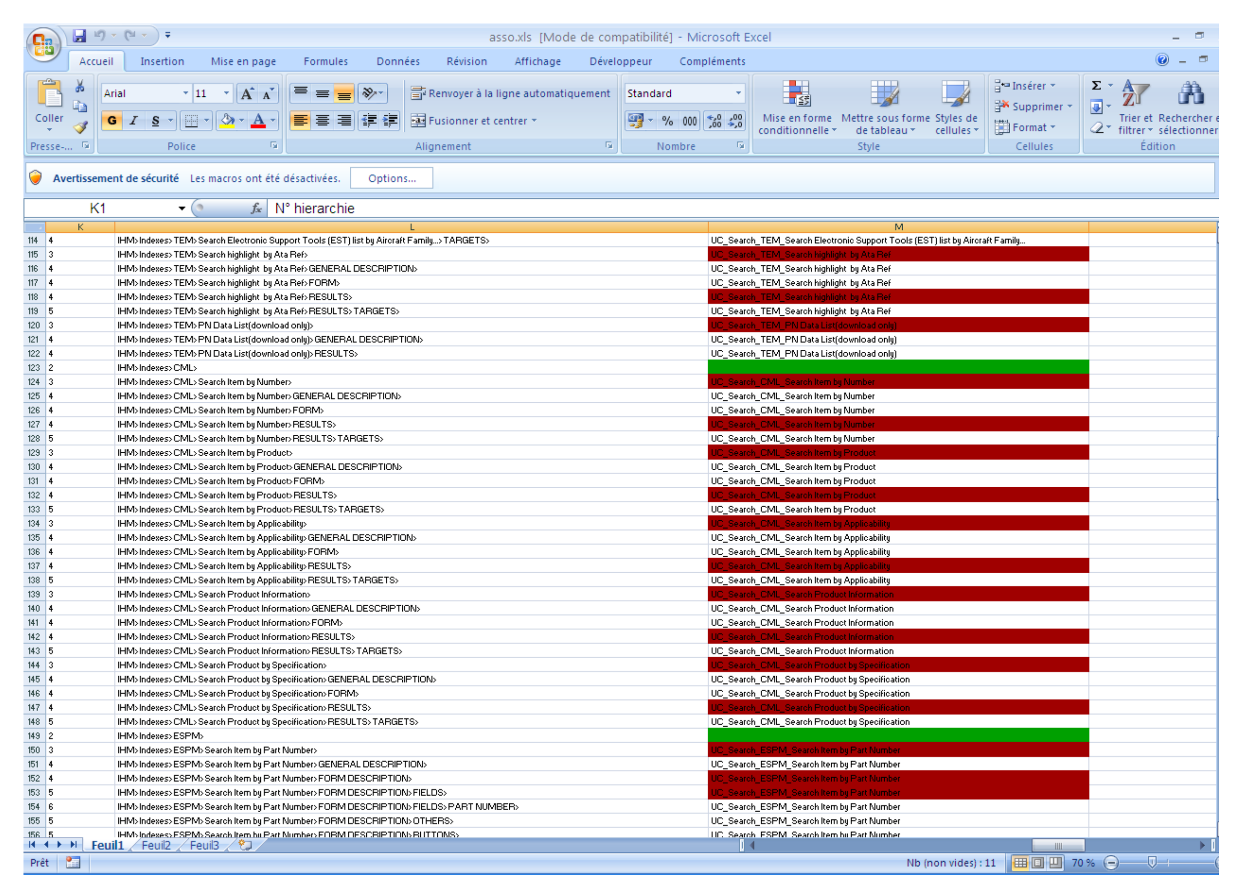
Pour toutes ces raisons là, l'automatisation des étapes du processus et l'étude de tous les projets pour proposer les règles de structure d'exigence commune sont nécessaires. C'est pourquoi l'étude du 2e scénario a été lancée.

Scénario 2 : Gérer les Use Cases hors de CaliberRM

Dans le scénario 2, les Use Cases sont générés automatiquement sous forme les Use Cases « vides » (qui ne contient que son nom). En se basant sur les exigences d'un projet, leur structure et leurs caractéristiques et en appliquant des règles proposées lors de l'analyse des projets, une liste de Use Cases associés aux exigences de ce projet sera créée. A partir de cette liste, une matrice de traçabilité entre les exigences du projet et ces Use Cases sera exportée sous forme d'un fichier consultable hors de CaliberRM (Word, Excel, pdf…).

Le fait que les Use Cases soient générés indépendamment du chef de produit et soient stockés dans un fichier hors de CaliberRM résout les problématiques expliquées avec les limites du scénario 1. Le chef de produit ne sera pas surchargé, la liste des nom des Use Cases en fichier sera partageable et utilisable par les autres personnes. Puisque la plupart des tâches du processus est effectuée hors de CaliberRM et HPQC, on peut réduire les risques de changement dans le BRD existant et aussi les erreurs venant de ce changement.

Par contre, il est normal que l'application des règles ne puissent pas créer les Use Cases qui couvrent la totalité des exigences d'un projet, certaines exigences n'ont pas la forme commune définie dans les règles. Le ratio de couverture dépend de chaque projet, mais doit être de plus que 80%. Les exigences restantes seront couvertes manuellement par le chef de produit. L'amélioration des règles sera faite de temps en temps pour augmenter le ratio de couverture et le nombre de projets applicables.



	K	L	M
114	4	IHM>Indexes>TEM>Search Electronic Support Tools (EST) list by Aircraft Family...>TARGETS>	UC_Search_TEM_Search Electronic Support Tools (EST) list by Aircraft Family...
115	3	IHM>Indexes>TEM>Search highlight by Ata Ref>	UC_Search_TEM_Search highlight by Ata Ref
116	4	IHM>Indexes>TEM>Search highlight by Ata Ref>GENERAL DESCRIPTION>	UC_Search_TEM_Search highlight by Ata Ref
117	4	IHM>Indexes>TEM>Search highlight by Ata Ref>FORM>	UC_Search_TEM_Search highlight by Ata Ref
118	4	IHM>Indexes>TEM>Search highlight by Ata Ref>RESULTS>	UC_Search_TEM_Search highlight by Ata Ref
119	5	IHM>Indexes>TEM>Search highlight by Ata Ref>RESULTS>TARGETS>	UC_Search_TEM_Search highlight by Ata Ref
120	3	IHM>Indexes>TEM>PN Data List(download only)>	UC_Search_TEM_PN Data List(download only)
121	4	IHM>Indexes>TEM>PN Data List(download only)>GENERAL DESCRIPTION>	UC_Search_TEM_PN Data List(download only)
122	4	IHM>Indexes>TEM>PN Data List(download only)>RESULTS>	UC_Search_TEM_PN Data List(download only)
123	2	IHM>Indexes>CML>	
124	3	IHM>Indexes>CML>Search Item by Number>	UC_Search_CML_Search Item by Number
125	4	IHM>Indexes>CML>Search Item by Number>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Number
126	4	IHM>Indexes>CML>Search Item by Number>FORM>	UC_Search_CML_Search Item by Number
127	4	IHM>Indexes>CML>Search Item by Number>RESULTS>	UC_Search_CML_Search Item by Number
128	5	IHM>Indexes>CML>Search Item by Number>RESULTS>TARGETS>	UC_Search_CML_Search Item by Number
129	3	IHM>Indexes>CML>Search Item by Product>	UC_Search_CML_Search Item by Product
130	4	IHM>Indexes>CML>Search Item by Product>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Product
131	4	IHM>Indexes>CML>Search Item by Product>FORM>	UC_Search_CML_Search Item by Product
132	4	IHM>Indexes>CML>Search Item by Product>RESULTS>	UC_Search_CML_Search Item by Product
133	5	IHM>Indexes>CML>Search Item by Product>RESULTS>TARGETS>	UC_Search_CML_Search Item by Product
134	3	IHM>Indexes>CML>Search Item by Applicability>	UC_Search_CML_Search Item by Applicability
135	4	IHM>Indexes>CML>Search Item by Applicability>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Applicability
136	4	IHM>Indexes>CML>Search Item by Applicability>FORM>	UC_Search_CML_Search Item by Applicability
137	4	IHM>Indexes>CML>Search Item by Applicability>RESULTS>	UC_Search_CML_Search Item by Applicability
138	5	IHM>Indexes>CML>Search Item by Applicability>RESULTS>TARGETS>	UC_Search_CML_Search Item by Applicability
139	3	IHM>Indexes>CML>Search Product Information>	UC_Search_CML_Search Product Information
140	4	IHM>Indexes>CML>Search Product Information>GENERAL DESCRIPTION>	UC_Search_CML_Search Product Information
141	4	IHM>Indexes>CML>Search Product Information>FORM>	UC_Search_CML_Search Product Information
142	4	IHM>Indexes>CML>Search Product Information>RESULTS>	UC_Search_CML_Search Product Information
143	5	IHM>Indexes>CML>Search Product Information>RESULTS>TARGETS>	UC_Search_CML_Search Product Information
144	3	IHM>Indexes>CML>Search Product by Specification>	UC_Search_CML_Search Product by Specification
145	4	IHM>Indexes>CML>Search Product by Specification>GENERAL DESCRIPTION>	UC_Search_CML_Search Product by Specification
146	4	IHM>Indexes>CML>Search Product by Specification>FORM>	UC_Search_CML_Search Product by Specification
147	4	IHM>Indexes>CML>Search Product by Specification>RESULTS>	UC_Search_CML_Search Product by Specification
148	5	IHM>Indexes>CML>Search Product by Specification>RESULTS>TARGETS>	UC_Search_CML_Search Product by Specification
149	2	IHM>Indexes>ESPM>	
150	3	IHM>Indexes>ESPM>Search Item by Part Number>	UC_Search_ESPM_Search Item by Part Number
151	4	IHM>Indexes>ESPM>Search Item by Part Number>GENERAL DESCRIPTION>	UC_Search_ESPM_Search Item by Part Number
152	4	IHM>Indexes>ESPM>Search Item by Part Number>FORM DESCRIPTION>	UC_Search_ESPM_Search Item by Part Number
153	5	IHM>Indexes>ESPM>Search Item by Part Number>FORM DESCRIPTION>FIELDS>	UC_Search_ESPM_Search Item by Part Number
154	6	IHM>Indexes>ESPM>Search Item by Part Number>FORM DESCRIPTION>FIELDS>PART NUMBER>	UC_Search_ESPM_Search Item by Part Number
155	5	IHM>Indexes>ESPM>Search Item by Part Number>FORM DESCRIPTION>OTHERS>	UC_Search_ESPM_Search Item by Part Number
156	5	IHM>Indexes>ESPM>Search Item by Part Number>FORM DESCRIPTION>BUTTONS>	UC_Search_ESPM_Search Item by Part Number

Figure 19 - L'exemple d'une matrice de traçabilité exportée sous forme d'un fichier Excel

Ce scénario a été appliqué dans le projet Associated Data V3. Après une analyse de l'opportunité et de faisabilité, ce scénario a été choisi pour suivre la conception et la réalisation du projet.

CONCEPTION ET RÉALISATION

Le choix du développement étant pris, il est à présent important d'analyser la solution à développer dans son ensemble c'est-à-dire d'examiner les choix organisationnels, d'architecture logicielle et d'environnements à établir.

SPECIFICATION/DOSSIER D'ANALYSE DES PROJETS

Cette étape est une étude avant tout développement. Elle permet de définir le contexte, plus technique, du projet. En effet, il est important de réfléchir sur les règles de base à utiliser et de conceptualiser la solution, avant même de commencer à programmer. Pour ce projet, l'étude sur la structure et les exigences des projets de SEVS est indispensable pour trouver une structure commune afin d'appliquer l'outil futur. Le but de cet étude est d'avoir un point de vue complèt sur les attributs des exigences, leur structure hiérarchique, de critiquer et de proposer les nouvelles règles de gestion de trois types généraux d'exigence : Searches, Links et Layout.

Durant le processus, plusieurs projets sont étudiés. Enfin, les 3 projets d'AirN@v choisis pour analyser sont Workshop V3, Associated Data V3 et Maintenance A350 car chacun représente un type de structure particulier.



Workshop V3

Le résultat de l'analyse de ce projet permet de construire des règles de structure de base et de définir 3 types d'exigence (Searches, Links, Layout).

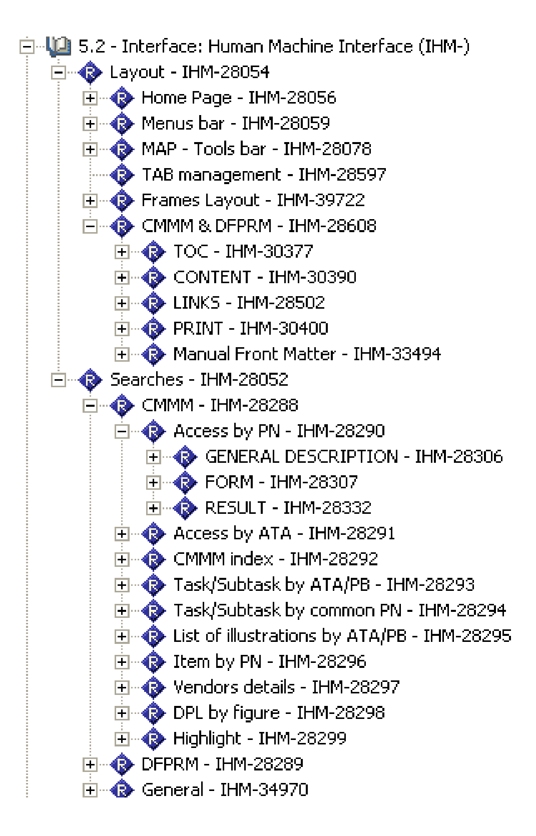


Figure 20 - Les exigences du projet Workshop

Couvrant l'A300, A310, A300-600, A318/A319/A320/A321, A330, A340, A380, Airn@v Workshop permet aux mécaniciens et les ingénieurs de consulter et de naviguer dans les documents principaux pourl'atelier: Component Maintenance Manual Airbus (CMMM) et Duct and Fuel Pipe Repair Manual (DFPRM).

En fait, les exigences de ce projet sont structurées de façon simple. Toutes sont mises dans le même type 5.2 – Interface : Human Machine Interface (IHM-). La liste des exigences est divisée en deux grandes parties : la partie « Searches » contient toutes les exigences des fonctions de recherche et la partie « Layout » contient toutes les autres exigences (celles de type layout, link… qui représente les demandes de l'interface et de lien de l'application).

Les exigences dans chaque partie sont encore divisées par manuel. Dans la partie « Searches », les exigences sont soit classées dans le manuel CMMM, soit dans DFPRM, soit dans General (les fonctions de recherche du menu principal, qui ne dépendent pas des manuels, comme Word Search…). Par contre, pour la partie Layout, les exigences sont divisées en élément d'interface (Home Page, Menus Bar, Tab, Frames…), qui sont indépendants des manuels, et en élément liés à chaque manuel (le TOC, Content, Links… affiché pour chaque manuel).

Donc la structure des exigences de ce projet est :

Searches :Searches>[Manuel]>[Nom de la fonction]>[Access/Form/Result/Target]

Links : Layout>CMMM&DFPRM>Links>[Nom du lien]>…

Layout : Layout>[Nom de l'élément de l'interface]>…

Les points faibles de cette structure hiérarchique viennent des exigences de liens. Puisqu'elles sont placé au même endroit pour tous les manuels (CMMM&DFPRM), la séparation des exigences selon leur manuel pour générer les Use Cases sera compliquée, surtout si chaque manuel dispose lui-même des différents liens à côté des liens communs. De plus, on risque de confondre des exigences de liens avec les exigences d'interface car elles sont placées au même endroit (Layout). Il vaut mieux placer les exigences de liens dans un endroit particulier et indépendant des autres types d'exigence. Par contre, la structure des exigences de Searches est parfaite et claire, donc elle sera prise en compte.

MAINTENANCE A350

Le résultat de l'analyse de ce projet permet d'étudier de manière approfondie l'opportunité et la faisabilité du 1er scénario.

Airn@v Maintenance A350 est un outil de consultation des documents de maintenance de l'A350. Puisque l'A350 n'est pas encore mise en service et livré au client (la date prévue de la première livraison de cet avion est en fin 2013), l'application Airn@v Maintenance A350 est en cours de développement.

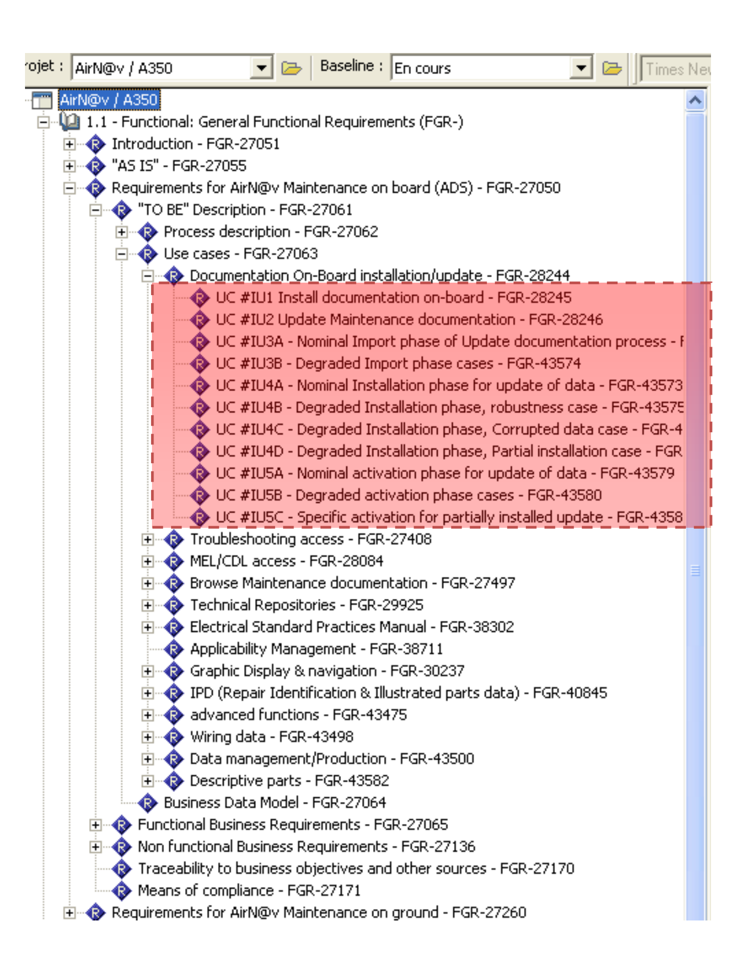


Figure 21- Les exigences du projet A350 (en rouge : les Use Cases générés par le chef de produit)

Lancé en Décembre 2008 et géré par M Jean BOUSQUET, c'est un gros projet avec plus que 1600 exigences pour le moment. Le BRD sera livré au SBA vers la fin du mois D'Août 2011. Les Use Cases de ce projet ne sont pas générés sur HPQC. Puisque c'est un nouveau projet, on peut tester directement les nouvelles règles de structure des exigences pour construire la traçabilité entre les Use Cases et les exigences.



Les Use Cases correspondants sont rédigés directement dans le BRD, dans un endroit particulier (la partie en rouge sur la figure). Le lien entre une Use Case et une exigence qui le correspond est fait en utilisant la fonction de traçabilité vers une autre exigence. A partir de ces liens de traçabilité, CaliberRM va fournir une matrice de traçabilité. Les règles appliquées sont celles du 1er scénario présenté dans la partie *Etude des différents scénarios, Scénario 1 (p.33)*.

Le grand nombre d'exigences de ce projet rend la structure hiérarchique des exigences compliquée. Toutes les exigences d'interface sont mises dans un même type particulier (5.2 Human Machine Interface : IHM-), les autres exigences (de type Searches et Links) sont mises dans le type 1.2 Functional : Data Management (FDM-). Dû au nouveau fonctionnement de l'Airn@v A350, les exigences « Searches » sont inclus dans les exigences « Links », les fonctions de recherche sont considérées aussi comme un lien entre les manuels. Cette structure induit le même problème au niveau de la confrontation ces deux types de exigences. De plus, les exigences-filles « Access mode» des exigences de type « Searches » ne sont pas situées dans le même endroit avec les autres exigences-filles de ce type (Result/Form/Target) de même les exigences ne sont pas classées selon leur manuel… Pour résoudre ces difficultés, il faut utiliser plusieurs règles pour reconstruire la structure des exigences et cela peut prendre du temps.

ASSOCIATED DATA V3

L'analyse de ce projet permet d'ajouter les contraintes de construction des exigences dans le but de perfectionner les règles afin de rendre le processus de génération des Use Cases plus efficace.

Disponible pour tous les programmes Airbus, AirN@v / Associated Data est un outil de consultation pour les manuels Consumable Material List (CML), Standards Manual (SM), Electrical Standard Practices Manual (ESPM) et Tool & Equipment Manual (TEM) incluant également les informations de Tool Equipment Index (TEI) et Support Equipment Summary (SES). AirN@v / Associated Data est nécessaire pour la maintenance et la réparation des avions Airbus.

La structure des exigences dans ce projet est cohérente avec les règles. Toutes les exigences sont placées sous un même type (5.2 Human Machine Interface IHM-). Les exigences de types différents sont mises dans des endroits différents (« Indexes » pour les exigences de Searches, « Links » pour les exigences de Links, « Layout pour les exigences de « Layout »). Ensuite, l'exigence du niveau hiérarchique suivant est toujours le nom du manuel pour lequel s'applique l'exigence (SM, CML, TEM,…).

Donc les exigences ont presque la même forme d'hiérarchie :

Searches : IHM>Indexes>[Nom Manuel]>[Nom Search]>[Description/Form/Result/Target]

Links : IHM>Links>[Nom Manuel]>[Internal/External]>[Nom Link]

Layout: IHM >Layout>[Nom Manuel]>[Element de Layout]

Donc on peut en déduire une forme commune pour toutes les exigences :

IHM>[Type d'exigence]> [Nom Manuel]>…

Avec ce « type d'exigence » qui vaut soit Searches, soit Links, soit Layout… La suite de la forme dépend de chaque type d'exigence.

Cette forme jouera un rôle principal dans l'algorithme de la génération des Use Cases correspondants aux exigences qu'on développera après.



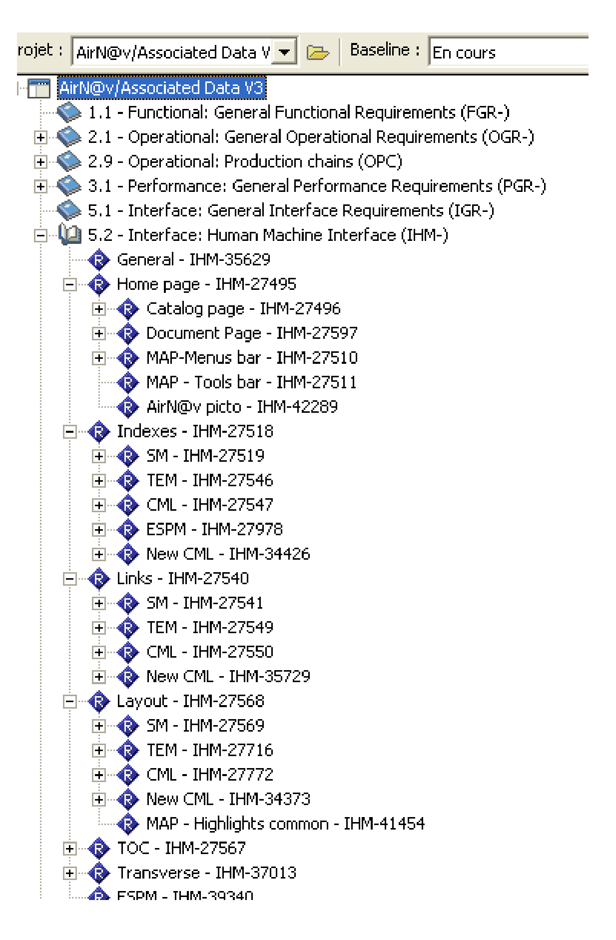


Figure 22 - Les exigences du projet Associated Data

Choix de langage de développement

Suivant les demandes d'Airbus (les fichiers résultats doivent être en format Excel pour adapter aux autres processus du service) et les outils existants fournis par Airbus, 3 solutions ont été analysées en utilisant 3 méthodes de codage différentes, dont chacun a ses propres avantages et inconvénients et qui sont plus ou moins hérité de l'autre. La conception n'a pas vraiment changée, mais le choix de logiciel de codage a changé suivant les différents scénarios. La première étape (qui utilise avec Document Factory pour générer le fichier Word) reste la même car imposée par l'utilisation de l'outil CaliberRM, pourtant les étapes suivantes peuvent être développées en code Excel, en VBA ou en Batch Code.

Solution 1 : Code Excel

La première solution est étudiée pour le cas où la mise en place de l'application est urgente. Le développement de l'outil ne doit pas durer trop longtemps et doit rester simple. C'est pourquoi, cette solution utilise directement les codes de base fournis par Excel, c'est-à-dire décrit directement les formules dans les cellules du fichier Excel pour avoir les résultats.

Le « code » Excel, en général, est constitué par les fonctions prédéfinies dans Excel. Elles sont utilisées pour construire directement les « formules » dans les cellules pour effectuer les calculs sur les autres cellules. Ses fonctions sont souvent destinées aux utilisateurs non-informaticiens grâce à leur principe simple.



Les codes pour écrire les formules sont simples, facile à développer. En plus, puisque ces formules sont exécutées directement dans chaque cellule du fichier, le temps d'exécution est négligeable. L'exécution directe du code diminue aussi le temps pour faire les Test et détecter les bugs potentiels.

Pourtant, il faut lancer ces formules manuellement et cela peut poser des problèmes avec les utilisateurs qui ne connaissent pas encore Excel. Ensuite, le fonctionnement des formules des cellules d'Excel est très limité car elles sont utilisées pour les calculs simples et directs. On ne peut implémenter que des fonctions très simples avec ces formules, le nombre de caractères de code d'une cellule est limité, donc le développement de procédures est impossible. Ces limites posent un gros problème pour l'évolution de l'outil après la mise en service.

J'ai commencé le développement de l'outil avec ce « langage », car cela est simple et ne prend pas beaucoup de temps pour finir (en fait juste une semaine, 5 jours, pour tout coder et fixer les bugs). Ce premier essai m'a permis de mieux analyser et estimer la faisabilité du projet et d'avoir un vue plus complète sur l'outil. J'ai arrêté de suivre ce codage après avoir réalisé ses limites d'évolution.

SOLUTION 2 : CODE VBA

Comme on l'a analysé, la première solution porte en elle-même beaucoup d'inconvénients, surtout sur les limites d'évolution du code. L'exigence posée est de trouver une solution qui permette de développer et de faire évoluer l'outil en plusieurs spécifications possibles. Pour satisfaire cette demande, il faut trouver un langage plus « évolué », c'est-à-dire plus statique et plus structuré, donc le VBA peut être choisi.

Cette solution utilise le code VBA (Visual Basic for Applications, une implémentation de Microsoft Visual Basic incluse dans toutes les applications de Microsoft Office) dans Excel pour créer des macros et des formulaires. L'outil est développé en procédures, chaque procédure réalise une étape ou une partie d'une étape de la spécification. L'utilisateur lance l'outil en lançant ces macros dans le fichier Excel ou en saisissant dans les formulaires.

Microsoft Visual Basic est un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, donc le développement en VBA n'est plus limité comme celui avec les formules d'Excel. L'évolution de l'outil est aussi facilitée car on a pu diviser l'outil en plusieurs procédures, ce qui facilite beaucoup la spécification, la planification et le codage. Les formulaires de VBA sont aussi intéressants pour implémenter une simple interface homme-machine avec les formulaires sous forme des fenêtres et objets Windows qui sont déjà habitués par l'utilisateur. En résumé, l'avantage principal de ce scénario est la non-limitation des fonctionnalités qu'on peut développer avec ce langage.

Pourtant, ce langage de codage a aussi ses points faibles. Premièrement, le temps d'exécution est long. Cela pose de gros problèmes dans l'utilisation de l'outil et surtout dans les phases de Test car on perd de temps chaque fois qu'on lance l'outil. Les tests sur les différents projets de SEVS donnent un résultat inattendu : il faut 60-90 minutes pour les projets ayant plus que 1500 exigences. Un autre problème de ce scénario est le fait qu'on ne peut pas totalement automatiser l'outil. Ce problème a une relation directe avec le problème du temps d'exécution, car l'augmentation du « taux » de l'automatisation amène l'augmentation du temps d'exécution (il y a certaines tâches qu'on fait plus vite de façon manuelle que de façon automatique, c'est-à-dire faire par l'ordinateur).

En dépit des points faibles listés ci-dessus, j'ai enfin décidé de l'utiliser pour profiter de la dynamique du langage VBA. Les problèmes seront résolus dans la partie de réalisation de l'outil.



SOLUTION 3 : VBA & BATCH CODE

Il s'agit d'une solution optimisée, qui répond à des exigences posées par l'analyse préalable du projet (le cahier des charges) et aussi des questions d'augmentation de performance, d'automatisation de toutes étapes de l'outil, de la facilité d'utilisation et d'installation… Le Batch Code est utilisé pour répondre à ces demandes.

Un Batch file (**.BAT)** est l'extension d'un fichier de commandes MS-DOS qui permet de concevoir des scripts qui seront interprétés par le "shell" ou interpréteur de commandes (command.com ou cmd.exe) pour notamment exécuter des fichiers .EXE ou .COM. Cette extension est principalement utilisée sur les systèmes d'exploitation de Microsoft (DOS et Windows). Elle peut être assimilée (dans une certaine mesure) à l'extension **.sh** des scripts shell Unix (ceux du Bourne shell plus exactement).

Cette solution est en fait une extension de la 2e solution afin d'améliorer l'efficacité de l'outil, surtout pour rendre l'outil plus simple à installer et à utiliser. Le but est de diminuer au plus petit nombre possible les actions de lancement de l'outil (clique ou saisi de texte). Ce scénario supprime aussi les étapes intermédiaires manuelles du transfert du fichier Word au fichier Excel, il rend automatique toutes les étapes.

```
C:\WINNT\system32\cmd.exe
c:\Users\T0101313\My Documents\VBA Doc\Asso data\test_02\batch code\Completed fo
lder>echo off
 9:51:21,32 [UC generator] generating Word file by Document Factory
 9:58:19,53 [UC generator] DONE
 9:58:19,53 [UC generator] word file generated
 9:58:19,53 [UC generator] generating use cases on excel file
 9:58:19,53 [UC generator] activating macro excel...
 9:59:48,50 [UC generator] DONE
 9:59:48,50 [UC generator] result file created
c:\Users\T0101313\My Documents\VBA Doc\Asso data\test_02\batch code\Completed fo
lder\~result.doc
Le processus ne peut pas accéder au fichier car ce fichier est utilisé par un au
tre processus.
Appuyez sur une touche pour continuer... _
```

Figure 23 - Le fenêtre de commande de l'outil

Les problèmes du code Batch sont les difficultés dans la compatibilité et l'applicabilité du Batch Code dans Word et Excel. En plus, le manque des documents de ce langage (car ce langage n'est pas un langage courant et souvent utilisé) m'empêche aussi de le suivre.

Comme cette solution est arrivée quand j'ai eu presque fini la spécification détaillée en VBA, elle hérite de tous ce qui est compris dans le 2e scénario. C'est pourquoi elle est juste une extension de la 2e solution, on ne peut pas l'implémenter avant de finir le 2e scénario.

C'est pourquoi j'ai décidé de prendre en compte cette solution dans l'évolution de l'outil.



On peut résumer les 3 solutions dans la table ci-dessous :

Tableau 1 - Tableau de comparaison des 3 solutions de développement

Solution	Quasi-manuel	Quasi-auto	Total-auto
Langage utilisé	Excel code	Visual Basic	VBA + Batch Code
Ratio d'automatisation	20%	80%	100%
Performance	Faible	Moyen	Fort
Temps d'exécution	<1 min	15-60'	10-40'
Facilité d'utilisation	Faible	Moyen	Très simple
Développement	Simple	Moyen	Compliqué
Evolution	Très limité	Moyen	Fort

EXPORTER DU BRD AVEC DOCUMENT FACTORY

DES TEMPLATES DE DOCUMENT FACTORY

Comme c'était indiqué dans la partie *Applications existants, Borland CaliberRM (p.26),* l'outil Document Factory de CaliberRM est utilisé pour la première étape du processus : exporter la liste de toutes les exigences du projet en un fichier Word afin d'y appliquer les règles pour générer les Use Cases.

La première étape de la génération d'un document avec Document Factory consiste à créer un modèle Microsoft Word. Un modèle contient les données imprimables de toutes sortes (y compris en-têtes de colonne, libellés et numéros de page) que l'on souhaite intégrer au document. Un modèle contient également des commandes, des mots-clés et des opérateurs Document Factory.

Les commandes contrôlent la sélection, l'ordre et le regroupement des exigences dans le document.

Les mots-clés sont des marques de réservation qui correspondent à des informations relatives au projet et aux exigences. Ils indiquent les données à intégrer au document. Les mots-clés doivent figurer dans le modèle dans l'ordre exact où ils apparaissent dans le gestionnaire d'exigences.

Les opérateurs nous permettent d'appliquer des critères très précis lors de la recherche de groupes d'exigences.

Document Factory utilise un jeu de commandes spéciales appelées $Commandes et destinées à remplir et à formater le document. Les **$Commandes** ne sont pas imprimables et n'apparaissent pas dans le document formaté. Certaines commandes, comme **$Begin_list/$End_list**, servent à traiter les listes d'informations, telles que les listes de liens de traçabilité ou les listes d'utilisateurs responsables d'une exigence. Utilisées seules, les $Commandes n'insèrent pas de données dans le document. Elles doivent être utilisées conjointement à des mots-clés.



LE TEMPLATE « TEMPLATE.DOT »

Le template « template.dot » utilisé pour exporter les exigences a la forme suivante :

Req Id	Standard	Origin	TRACES	Requirement Name	N° hierarchy	ParentId	Id
$BEGIN_SECTION $SORT{ type,type,hierarchy }<<tag>><<id_number>>	<<Product Version>>	<<origin>>	$begin_traces(tm_traces, Direct_Traces, All_Objects)***<<Trace_ReqTag>> - <<trace_direction>>*** $end_traces	<<name>>	<<hierarchy>>	<<parentId>>	<<id_number>>

$END_SECTION

En se basant sur ce template, Document Factory va exporter un fichier contenant les informations de toutes les exigences du projet choisi :

- **Req Id** (Requirement Id) : l'identifiant de l'exigence, souvent sous forme ***<type de requirement>-<numero id>***. Par exemple : on a **IHM-27565, FOB-34928** d'où IHM, FOB sont des types d'exigence, 27565 et 34928 sont des identifiants des exigences. Dans un projet, l'identifiant d'une exigence est unique par contre on peut avoir des exigences de même type.

- **Product Version** : chaque exigence est sous contrôle de version et les utilisateurs peuvent consulter les versions précédentes d'une exigence et comparer ces versions, cet attribut représente la version du produit de l'exigence . Par exemple les versions des exigences de Workshop V3 sont souvent **V3.1, V3.1.1, V3.0**)

- **Origin** : l'origine (si elle existe) de l'exigence. Certaine exigence hérite une autre exigence et a l'origine d'une exigence d'un autre projet. Ces exigences sont souvent les exigences communes pour des applications proches. En fait, les applications de la famille AirN@v ont certaines exigences communes stocké dans un BRD appelant Airn@v Common, les exigences communes dans chaque projet ont l'origine de ce BRD. Par exemple, l'exigence **IHM-33500** du projet A350 a l'origine d' **AirN@v Common V3**.

- **Traces** : La traçabilité est essentielle à l'évaluation de la couverture des exigences et permet d'estimer l'impact des modifications d'exigences. Cet attribut est représenté sous forme <req id de l'exigence tracée>-<direction de la traçabilité>. Par exemple : la trace de l'exigence **IHM-33500** est **IHM-18603 - DEPUIS IHM-14249 –DEPUIS**, cela signifie que cette exigence a liens de traçabilité, ces deux liens sont venus des exigences **IHM-18603** et **IHM-14249**. L'utilisation de cet attribut est à discuter, il est indispensable dans le scénario 1 mais dans le scénario 2, son rôle n'est plus important que les autres attributs.

- **Requirement Name** : le nom (titre) de l'exigence. (ex : **Search Panel Information**). Cet attribut joue un rôle important (comme un composant) dans la construction du chemin d'arborescence (le chemin hiérarchique) d'une exigence ainsi que la génération des Use Cases. Un mauvais nom d'une exigence pourrait donner les mauvais résultats du processus. C'est pourquoi le nom d'une exigence doit respecter strictement des contraintes proposées.



- **N° hierarchy** : ce numéro représente la position de l'exigence dans l'arborescence des exigences et sert à distinguer les exigences parent et enfant. Le numéro de hiérarchie de l'exigence est composé par une série de nombre, séparée par un point. Le numéro d'éléments (de nombres) de cette série indique le niveau hiérarchique de l'exigence. Les exigences du plus haut niveau hiérarchique ont le numéro hiérarchique composé par un nombre (**1, 2, 3, 4**), elles sont surtout les exigences de type, qui indique le type de toutes ses exigences « filles » (comme **FOB, IHM, FRD**…). Le numéro hiérarchique d'une exigence « fille » est construite en prenant directement le numéro hiérarchique de sa mère et ajoutant un numéro à la fin de la série indiquant l'ordre de l'exigence « fille » parmi les autres « filles » de sa mère. Par exemple : une exigence de numéro d'hiérarchie **1.4** a 3 filles, donc ses filles ont les numéros d'hiérarchie successivement **1.4.1**, **1.4.2** et **1.4.3** selon l'ordre dans l'arborescence de leur mère. Donc, en connaissant le numéro hiérarchique d'une exigence, on peut trouver son exigence mère.

- **ParentId** : Cet attribut n'est pas un attribut initialement créé par CaliberRM, il a été ajouté particulièrement dans la version de CaliberRM d'Airbus. Cet attribut est indispensable dans les deux scénarios. Il joue le rôle principal dans la construction des chemins, ce qui sera utilisé après pour la génération des Use Cases. C'est pourquoi, cet attribut est préalable pour le processus.

  ParentId est composé par plusieurs tag, qui donnent les informations importantes. Sa forme est de ***<identifiant exigence « mère »>-<type classification>-<niveau hiérarchique>.***

- **Identifiant exigence « mère »** : l'identifiant de l'exigence mère de l'exigence courante, a la même forme qu'un identifiant d'une exigence normale. Les exigences de « type » (exigences au niveau hiérarchique le plus haut et qui n'ont pas de mère) n'ont pas cet attribut dans ParentId.

- **Type de classification** : cet attribut indique le type de classification de l'exigence. Il y a 2 types de classification : table et titre. Une exigence de type « table » est une exigence « fille » pure, c'est-à-dire elle n'a aucune fille au-dessous d'elle. Par contre, une exigence de type « titre » est une exigence « mère », elle doit avoir au moins une exigence « fille ». Cet attribut est très utile dans l'algorithme de la construction du chemin d'arborescence d'une exigence et même la génération des Use Cases car il aide à déterminer les exigences « fille » pure, à partir de celles-là, on peut déduire le chemin hiérarchique ou affecter directement le nom du Use Case correspondants ('algorithme sera détaillé dans la partie *Construction du chemin d'arborescence, Algorithme de construction, p.43*).

- **Le niveau hiérarchique** : cet attribut est l'extrait de l'attribut N° hierarchy, en fait, il compte le numéro d'élément composant dans la série du numéro hiérarchique de l'exigence courante.

  A
  ├─ A1 (id=**14425**)
  │ └─ A2 (ParentId=**14425-table3**)

  <u>Exemple</u> : une exigence (A2) ayant le ParentId **14425-table3**, c'est-à-dire l'identifiant de sa mère est 14425. A2 est une exigence de type « table », c'est-à-dire elle n'a pas de fille, elle est une exigence « fille » pure. Son niveau d'hiérarchie est 3, c'est-à-dire son numéro hiérarchique se compose d'une série de 3 nombres (de forme **x.y.z**), elle a une exigence « mère » (A1) et sa mère a aussi son exigence « mère » (A) qui est une exigence de « type », l'exigence ayant le plus haut niveau hiérarchique.

- **id** : Identifiant d'une exigence (pas de type ajouté).

Exemple : Voici une partie du fichier Word généré par Document Factory en utilisant le template « template.dot » et qui contient les exigences du projet Associated Data V3.

Tableau 2 - Partie du fichier BRD de Associated Data V3

IHM-27517	V3.1			content	2.1.1	27496-table3	27517
IHM-27597	V3.1			Document Page	2.2	27495-titre2	27597
IHM-27599	V3.1			Floating windows	2.2.1	27597-table3	27599
IHM-27598	V3.1			Frames disposition	2.2.2	27597-titre3	27598
IHM-27889	V3.1			specific ESPM frame	2.2.2.1	27598-table4	27889
IHM-27510	V3.1			MAP-Menus bar	2.3	27495-titre2	27510
IHM-27565	V3.1			SM	2.3.1	27510-table3	27565
IHM-27600	V3.1			TEM	2.3.2	27510-table3	27600
IHM-27601	V3.0			CML	2.3.3	27510-table3	27601
IHM-37268	V3.1			New CML	2.3.4	27510-table3	37268

TRANSPORTER LES EXIGENCES DU FICHIER WORD AU FICHIER EXCEL

C'est une étape intermédiaire, le but est de transporter le contenu du fichier Word qu'on a créé vers un fichier Excel pour les traitements suivants. Comme le langage de codage choisi est le VBA s'appliquant sur Excel, toutes les données doivent être passées à un fichier Excel afin de pouvoir appliquer les macros développées en code VBA. Cette étape va réaliser le fait de copier tout le tableau des exigences du fichier Word dans les cellules d'un fichier Excel. Le fichier Excel a déjà des macros développé antérieurement. Une fois toutes les données copiées dans les cellules, il faut juste lancer les macros pour générer les Use Cases et effectuer la couverture.

Deux façons ont été proposées pour réaliser cette étape : une façon automatique et une façon manuelle, chacune a ses propres points forts et faibles.

FAÇON AUTOMATIQUE

Il suffit d'utiliser le mode « Auto » de l'outil. La copie du contenu du fichier Word vers le fichier Excel est faite automatiquement à l'aide des macros développées au sein de ce fichier Excel. Quand le fichier « UC_generator.xls » est ouvert par l'outil, une fenêtre de mode « Auto » sera affichée. L'utilisateur ne doit que appuyer sur le bouton « OK » pour sélectionner l'endroit pour sauvegarder le fichier final et le nom de ce fichier. Dès que la commande est lancée, le processus va être lancé sans arrêt jusqu'à la fin, c'est-à-dire la génération des Use Cases, d'un log… Les dossiers finaux (la matrice de traçabilité et le fichier log) sont aussi créés automatiquement.

Cette méthode est plus idéale pour les utilisateurs non-informaticiens, elle se rapproche l'automatisation de la totalité du processus. L'interface du fichier Excel est aussi simple, comparable à celles des autres applications de Windows que les utilisateurs ont déjà l'habitude d'utiliser. En suite, l'utilisateur n'a pas besoin de lancer lui-même les macros car elles vont se lancer automatiquement. Cela peut aussi prévenir les erreurs venant du lancement manuel des macros (confondre de macros, de feuilles ou erreurs de paramètres…). Les fichiers de résultat sont créés automatiquement, ce qui peut réduire le temps et le travail de l'utilisateur.



Par contre, son gros inconvénient vient du temps d'exécution de l'outil. Le fait de copier le contenu d'un tableau d'un fichier Word dans les cellules d'un fichier Excel peut prendre beaucoup de temps car l'outil va copier chaque case du tableau Word dans une cellule d'Excel. Les tests ont montré que pour les tableaux ayant moins de 500 lignes (donc 500 exigences), le temps d'exécution de l'outil est moins de 10 minutes. Par contre, pour les projets ayant plus que 1500 exigences, il faut au moins une heure pour finir tout le processus (le test sur le projet A380 ayant 1638 exigences (donc 1638 x 8 ~ 13 000 cellules copiées) a pris 73 minutes donc 1 heure et 13 minutes de temps d'exécution).

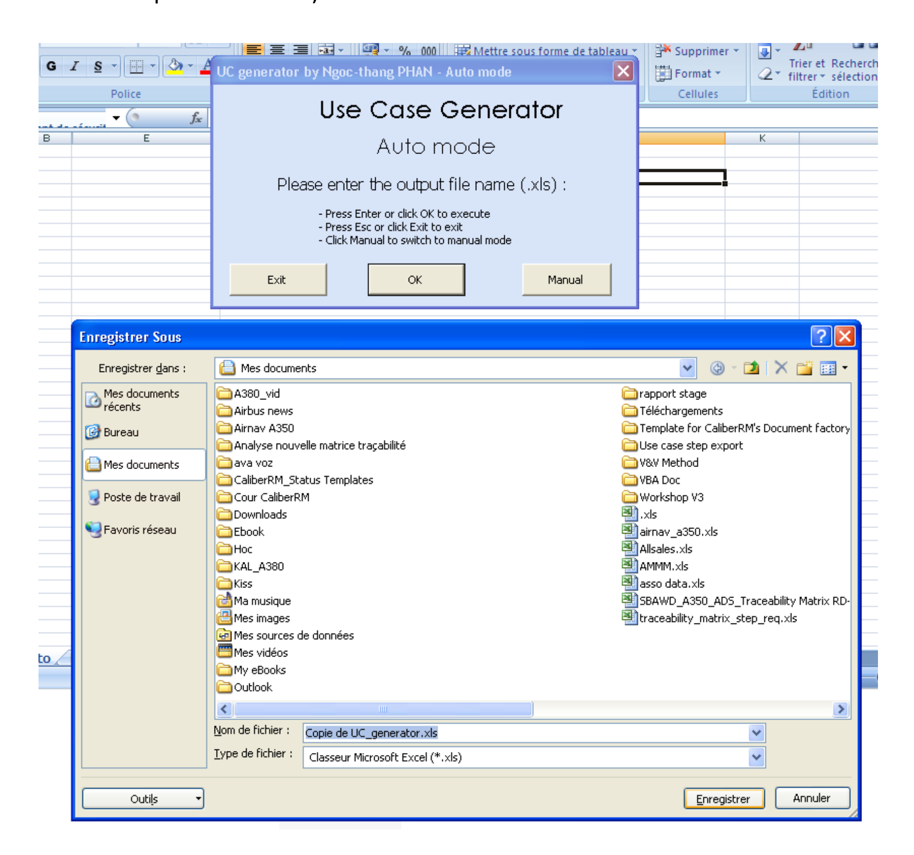


Figure 24 - Fenêtre du mode automatique

Durant l'exécution d'une macro d'Excel, tous les autres traitements sur les autres fichiers Excel sont bloqués, c'est-à-dire, dans ce temps là, on ne peut rien faire sur les autres fichiers Excel. Cela peut poser des problèmes si on a plusieurs fichiers Excel en cour d'utilisation simultanée.

En réalité, cet inconvénient a déjà posé pas mal de difficulté dans les tests. A cause de son temps d'exécution potentiellement long, les tests sur cette méthode peuvent aussi durer plus longtemps que prévu.

FAÇON MANUELLE

La deuxième méthode est l'inverse de la première. Le fait de copier le contenu du fichier Word dans le fichier Excel est effectué manuellement par l'utilisateur. Il doit juste changer en mode « Manual » et utiliser le copier-coller normal pour copier tout le contenu du tableau Word et le coller dans la feuille « Manual » du fichier « UC_generator.xls ». Cette feuille contient déjà les macros développées antérieurement.

Une fois toutes les données passées dans la feuille « Manual » du fichier « UC_generator.xls », l'utilisateur n'a qu'à lancer la macro « feuille2.main » de ce fichier en appuyant le bouton affiché sur la feuille. Après le lancement de cette macro, les autres macros seront exécutées successivement et on n'a pas besoin de les lancer. Après la génération des Use Cases, une fenêtre sera affichée pour que l'utilisateur puisse saisir le nom du fichier résultat qui va être créée à la fin du processus.

Cette méthode a résolu le problème du temps d'exécution de la méthode automatique. Puisque le fait de copier les données d'un fichier Word vers un fichier Excel est l'étape qui dure le plus longtemps (donc elle prend la plupart du temps d'exécution de l'outil), il faut juste éliminer cette étape dans le processus de l'automatisation. Le fait de copier-coller manuellement va réduire beaucoup le temps total d'exécution, car les étapes restantes ne durent que 5-10 minutes pour n'importe quel projet (même moins de 10 minutes pour les gros projets avec plus de 1500 exigences). Donc avec cette méthode, on ne perd pas beaucoup de temps pour l'exécution de l'outil.

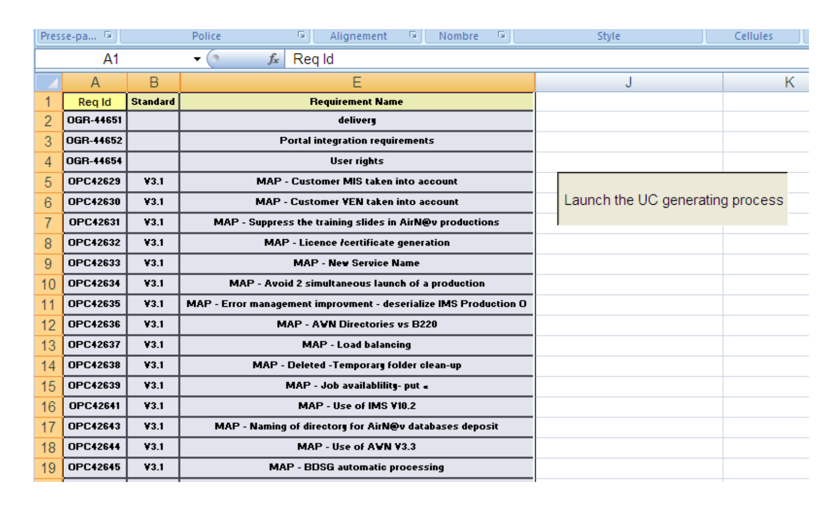


Figure 25 - Lancement de la génération des UC

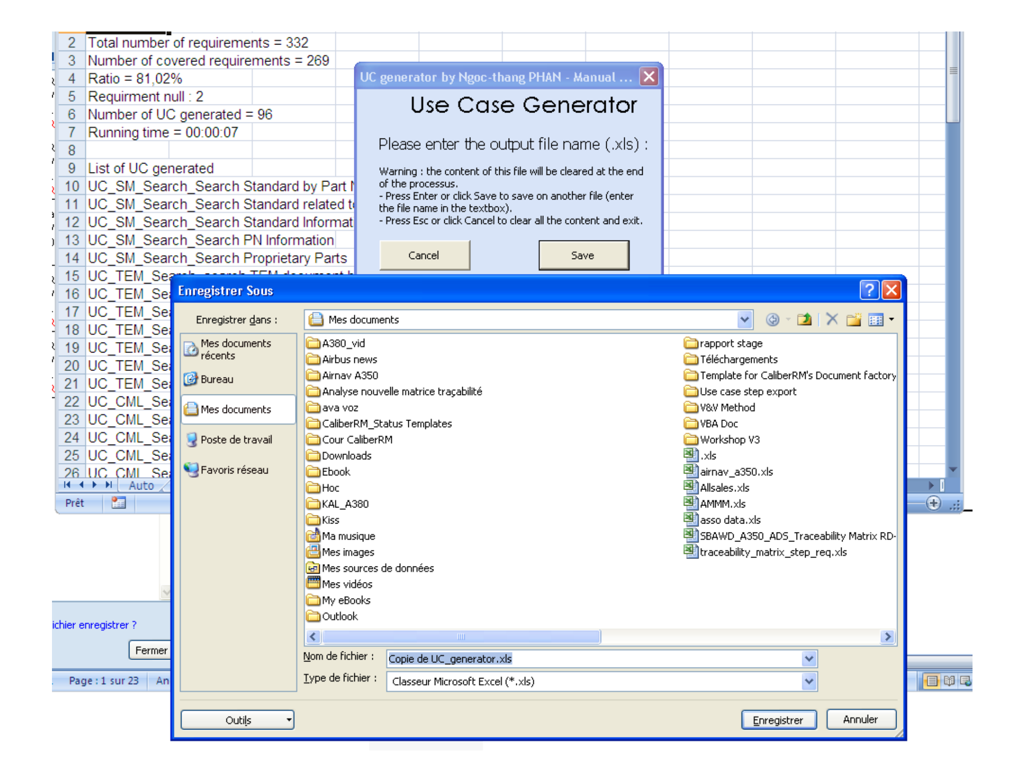


Figure 26 - Fenêtre du mode Manuel

Pourtant, comme cette méthode est l'inverse de la méthode automatique, ses problèmes viennent des manipulations manuelles à certaines des étapes du processus. Le fait de copier-coller à la main risque de produire des erreurs comme le manque de données, les fait de copier les mauvaises données, de se tromper de feuille d'Excel. Ces problèmes sont souvent rencontrés avec les gros projets, où le nombre des données est important. Ensuite, pour les utilisateurs non-informaticiens, surtout les personnes qui ne travaillent pas



souvent avec Excel, il est un peu compliqué pour eux de réaliser les actions comme activer les macros et lancer les macros.

Pour conclure, cette méthode permet de réduire le temps d'exécution du processus, par contre, elle ajoute des tâches à l'utilisateur, à faire lui-même à la main.

L'usage de ces deux méthodes est en train d'être discuté. On peut utiliser ces deux méthodes de façon plus flexible : pour les gros projets, on utilise la méthode manuelle et pour les petits projets, on applique la méthode automatique. L'usage de ces deux méthodes dépendra du choix de l'utilisateur.

CONSTRUCTION DU CHEMIN D'ARBORESCENCE D'EXIGENCE

POURQUOI UN CHEMIN D'ARBORESCENCE

Généralement, une Baseline CaliberRM contient des ensembles d'exigences reliées par des relations hiérarchiques (mère-fille). Une fois réunies, ces exigences constituent une arborescence (comme les fichiers et les répertoires dans le système d'exploitation Windows). On appelle chemin d'arborescence d'une exigence, le chemin pour accéder à une exigence en partant de l'exigence « racine » et en traversant toutes ses exigences « mères » jusqu'à cette exigence.

Par exemple, avec l'arborescence suivante :

- Le chemin d'arborescence de l'exigence H est :

  A > G > H

- Le chemin d'arborescence de l'exigence E est :

  A > B > D > E

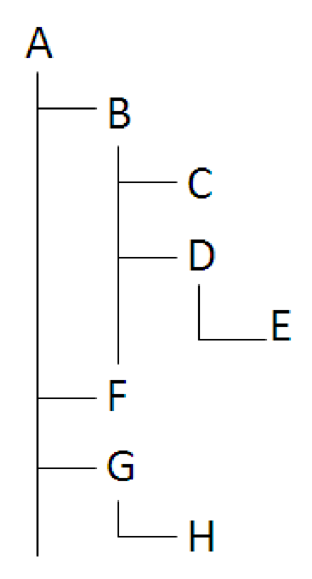


Figure 27 - Exemple d'arborescence des exigences

Les règles sont les mêmes que pour les fichiers ou les répertoires en Windows, B, F, G ont le même niveau hiérarchique et ont la même exigence mère, le cas de C et D est identique.

La construction du chemin d'arborescence des exigences est très important car une fois qu'on connait le chemin d'arborescence des exigences, on peut déduire la relation globale entre les exigences. Ensuite, comme les règles appliquées pour générer les Use Cases sont basées sur le chemin d'arborescence des exigences, le chemin d'arborescence d'une exigence peut influencer beaucoup sur le résultat de la couverture des Use Cases sur les exigences. Un mauvais chemin d'arborescence d'une exigence risque de bloquer non seulement la génération de son Use Case mais aussi des Use Cases de toutes ses exigences proches.

De plus, le chemin d'arborescence permet aussi au chef de produit d'avoir une vue complète sur le projet et de mieux gérer l'ensemble de ses exigences.

ALGORITHME DE CONSTRUCTION

Une fois toutes les exigences avec leurs attributs dans le fichier Excel, la construction du chemin d'arborescence peut être lancée.



Au début, pour chaque exigence, l'outil va extraire la partie qui représente l'identifiant de l'exigence mère. (Par exemple le ParentId d'une exigence est **27518-titre2** donc l'identifiant de l'exigence mère est **27518**)

Ensuite, l'outil va faire une simple boucle. Pour chaque exigence, en se basant sur l'identifiant de sa mère. Il va chercher l'exigence mère dans la liste les exigences du projet et récupérer le nom ainsi que le ParentId de cette exigence. L'outil va continuer de faire la même chose avec l'exigence mère trouvée et va se terminer en rencontrant une exigence de « type » (l'exigence n'ayant pas de « mère »).

Voici l'algorithme en utilisant un langage simple :

```
Tableau boolean trouve_pere[n]
Tableau exigence[n]
/*supposons que toutes les exigences sont stockées dans un tableau de dimension n*/
Pour i de 1 à n Faire
/*on balaye pour toutes les exigences*/
Debut
       /*initialise le chemin*/
Chemin <- « »
Tant que Id_pere(i) <> NULL Faire /*l'exigence a une mère*/
       Debut
              trouve_pere(i) <- FAUX
              Pour j de 2 à n Faire
              Debut
                     Si id(j) = id_pere(i) alors
                     /*si on trouve que la mere de l'exigence i est j*/
                            Resultat <- j
                            k <- j
                            trouve_pere(i) <- VRAI
                     Fin Si
              Fin Pour
              Si trouve_pere(i) = VRAI Alors
                     /*si on peut trouver la mère de l'exigence i*/
                     k <- resultat
                     /*ajouter le nom de l'exigence mère dans le chemin de la
fille*/
                     chemin <- chemin + nom_req(k)
              Sinon
                     /*si on ne peut pas trouver la mère de l'exigence i*/
                     Afficher « Parent not found »
                     Sortir de la boucle Tant que
              Fin Si
       Fin Tant que
       Si chemin <> NULL et trouve_pere(i) = VRAI alors afficher le chemin
Fin Pour
```

<u>Illustration :</u>

On veut construire le chemin d'arborescence de l'exigence « TARGETS » numéro 27812, qui a son ParentId 27807-table5. Au début, le chemin ne contient que le nom de cette exigence. A partir de son ParentId, on cherche dans la liste de toutes les exigences l'exigence ayant l'identifiant 27807, et on trouve « RESULTS ». On ajoute le nom de cette exigence dans le chemin d'arborescence de l'exigence « TARGETS ». En faisant la même chose avec l'exigence « RESULTS », on trouve sa mère « Search Product by Specification ». On ajoute son nom dans le chemin d'arborescence. On trouve « CML » et « Indexes » avec la même méthode. Comme « Indexes » n'a pas d'exigence mère, (son ParentId est « title1 » et son niveau hiérarchie est le plus haut =1), on arrête la boucle et la construction du chemin d'arborescence de l'exigence « TARGETS » est terminée.



Le chemin d'arborescence final de cette exigence est :

IHM>Indexes>SM>Search Standard Information>RESULTS>TARGETS>

Le tableau ci-dessous résume toutes les étapes de la boucle :

Etape	Requirement Name	Requirement Id	ParentId	Chemin
1	TARGETS	IHM-27812	27807-table5	TARGETS>
2	RESULTS	IHM-27807	27783-titre4	RESULTS>TARGETS>
3	Search Product by Specification	IHM-27783	27547-titre3	Search Product by Specification> RESULTS>TARGETS>
4	CML	IHM-27547	27518-titre2	CML>Search Product by Specification> RESULTS>TARGETS>
5	Indexes	IHM-27518	titre1 → STOP	Indexes>CML>Search Product by Specification> RESULTS>TARGETS>

Tableau 3 - Application de l'algorithme pour un cas précis

Génération des Use Cases

Une fois les chemins de toutes les exigences sont construits, la génération des Use Cases est lancée. Elle se base sur le chemin d'arborescence de chaque exigence et les règles de structure proposées de l'analyse préalable des projets.

Nom d'un Use Case

Le nom d'un Use Case a la forme : UC_<type de UC>_<nom manuel>_<nom du UC>

Où les composants sont :

- **Type de UC** : Après l'analyse de quelques projet, on peut déduire les types principaux des exigences:
  - **Search** : Use Case d'une fonction de recherche de l'application du projet. Ce Use Case couvre normalement des exigences de « access » (le chemin d'accès à la fonction de recherche), « form » (le formulaire de recherche à saisir par l'utilisateur), « result » (la formulaire de résultat de recherche) , « target » (les liens ou les cibles contenu dans la formulaire de résultat de recherche). Il y a deux types

de fonctions de recherche, fonction de recherche d'un manuel particulier ou fonction de recherche du menu principal.

Le nom d'un Use Case de type Search est souvent le nom de la fonction de recherche.

- **Link** : Use Cases d'un lien entre une source et une cible dans l'application. Ce lien peut être interne (qui relie à une autre partie du manuel) ou externe (qui relie à un autre manuel ou aussi une autre application).

  Le nom d'un Use Case de liens a souvent la forme : *<origine>_to_<destination>.*

- **Display** : Use Case d'un élément de l'interface (Catalogue Page, Table of Content, Font, Graphic Reference display…). Un Use Case de type Display va générer les différents Test Case pour chaque manuel. Ces exigences dans un projet sont souvent sous le répertoire « Layout ».

- **Print** : Use Case de la fonction d'impression.

- **Nom Manuel** : le nom du Manuel correspond au Use Case (par exemple CML, SM, TEM… du projet Associated Data V3).
- **Nom du UC** : le nom du Use Case, généré à l'aide du chemin d'arborescence de l'exigence et les règles.

PRINCIPE

L'analyse des projets permet de déduire une forme commune pour le chemin d'arborescence des exigences. Cette forme a été abordée dans la partie d'analyse du projet Associated Data (2.3.1.3.) :

IHM>[Type d'exigence]>[Nom Manuel]> [Nom d'exigence niveau 3] >[Nom d'exigence niveau 4]>…

En se basant sur cette forme, l'outil va en retirer les informations sur

- le type du Use Case : le composant ***Type d'exigence*** (exigence niveau 1)
- le nom du manuel : le composant ***Nom Manuel*** (exigence niveau 2)
- le nom du Use Case (nom du fonction de recherche, nom du lien ou nom d'élément de l'interface…) : le composant ***Nom d'exigence niveau 3***

Le reste du chemin d'arborescence (les exigences de niveau d'hiérarchie 4, 5 ou 6… s'il y a) n'est pas pris en compte pour composer le nom du Use Case, toutes les exigences ayant les mêmes exigences de niveau d'hiérarchie 1, 2, 3 (même type d'exigence, manuel et même exigence niveau 3) sont couvertes par le même UC.

Par exemple :

L'exigence TARGETS dont le chemin est le suivant :

IHM>Indexes>SM>Search Standard Information>RESULTS>TARGETS>

Elle est couverte par le Use Case : UC_Search_SM_Search Standard Information



D'autre part, l'exigence FORMS a le même type d'exigence, le même manuel et la même fonction de recherche (Search Standard Information) que l'exigence TARGETS :

IHM>Indexes>SM>Search Standard Information>FORMS>

Elle est aussi couverte par le même Use Case que la dernière : **UC_Search_SM_Search Standard Information**.

En fait, ces 2 exigences sont les composants « Form » et « Target » du Use Case de Search **UC_Search_SM_Search Standard Information**.

Règles de génération des Use Cases

Au début, il faut définir les règles appliquées pour la génération des Use Cases. Les Use Cases correspondants aux exigences doivent être strictement générés dans l'ordre de ces règles. L'explication de l'utilisation de ces règles sera détaillée après. Ces règles sont :

1) Pour toutes les exigences dont le chemin d'arborescence n'a pas été construit (ParentId non trouvé par exemple) on ne pourra pas déduire les Use Cases correspondants.

2) Pour toutes exigences de type « table », c'est-à-dire les exigences « filles » pures (qui n'ont pas de fille) ayant le niveau hiérarchique plus grand ou égal que 3 on déduira directement leurs Use Cases correspondants. Par contre, pour toutes les exigences de type « table » ayant le niveau hiérarchique plus petit que 3 on ne pourra pas déduire leurs Use Cases correspondants.

3) Il faut que le niveau hiérarchique d'une exigence soit supérieur ou égale que 3 pour pouvoir générer son Use Case correspondant parce que si son niveau hiérarchique est inférieur que 3, son chemin d'arborescence ne contient que les exigences du niveau 1 ou 2, et donc ne contient pas les 3 informations obligatoires pour construire le Use Case (type de l'exigence, nom du manuel, nom du Use Case), donc la génération du Use Case correspondant dans ce cas est impossible.

4) Pour une exigence de type « titre », c'est-à-dire l'exigence ayant un ou plusieurs exigences « filles »,on ne peut pas déduire son Use Case correspondant. Son Use Case doit dépendre de des Use Cases de ses filles :

   a) Si ses filles sont toutes couvertes par un même Use Case, alors cette exigence est aussi couverte par ce Use Case

   b) Si ses filles sont toutes couvertes mais par différents Use Cases, alors on ne peut pas déduire son Use Case correspondant mais cette exigence est aussi couverte.

   c) Si au moins une de ses filles n'est pas couverte par un Use Case, cette exigence n'est pas couverte non plus.

5) Pour certains types d'exigence on peut ne pas déduire leurs Use Cases correspondants. Par exemple : les exigences de type d'observations (FOB, NOB…) qui ne sont pas livrés au SBA.



DÉMARCHES

Au début, l'outil génère les Use Cases pour les exigences « filles » pures, car pour les exigences de ce type on peut déduire leurs Use Cases directement à l'aide des règles de génération des Use Cases. Ces exigences doivent avoir le niveau hiérarchique supérieur ou égal à 3.

A ce moment, il ne reste que les exigences « mères » (type « titre) et les exigences « filles » pures ayant le niveau hiérarchique inférieur à 3. Ces exigences « filles » pures sont négligées car selon les règles de générations, on ne peut pas générer les Use Cases pour elles.

Pour les exigences « mères », la génération des Use Cases n'est plus exactement une génération, aucun nouveau Use Case n'est généré. L'outil va juste affecter à ces exigences leurs Use Cases correspondants parmi les Use Cases de ces exigences « filles » en appliquant la 3e règle détaillée dans la partie précédente. L'algorithme de cette affectation est simplement une boucle qui balaye, pas dans le sens des exigences du plus haut au plus bas niveau hiérarchique, mais dans le sens inverse, en partant des exigences du plus bas niveau au plus haut niveau. Cette méthode va vérifier selon la règle 2 qu'on n'omet aucune exigence dans la liste : l'affectation de Use Case d'une exigence « mère » ne peut être réalisé qu'après l'affectation des Use Cases de toutes ses filles car d'après la règle 2, la génération du Use Case d'une exigence « mère » dépend des Use Cases de ses filles. La boucle est terminée quand toutes les exigences du plus haut niveau hiérarchique sont balayées.

Cas d'illustration :

On va voir les démarches de l'outil pour générer les Use Cases pour ces 10 exigences suivantes en appliquant l'algorithme ci-dessus.

N°	Chemin d'arborescence	Use Case		Type
1	IHM>Indexes>CML>		2	Mère
2	IHM>Indexes>CML>Search Item by Number>		3	Mère
3	IHM>Indexes>CML>Search Item by Number>GENERAL DESCRIPTION>		4	Fille
4	IHM>Indexes>CML>Search Item by Number>FORM>		4	Fille
5	IHM>Indexes>CML>Search Item by Number>RESULTS>		4	Mère
6	IHM>Indexes>CML>Search Item by Number>RESULTS>TARGETS>		5	Fille
7	IHM>Indexes>CML>Search Item by Product>		3	Mère
8	IHM>Indexes>CML>Search Item by Product>GENERAL DESCRIPTION>		4	Fille
9	IHM>Indexes>CML>Search Item by Product>FORM>		4	Fille
10	IHM>Indexes>CML>Search Item by Product>RESULTS>		4	Fille

L'outil commence par générer les Use Cases pour les exigences « fille » ayant le niveau hiérarchique supérieur ou égale à 3. Les exigences satisfaits à ce critère sont celles n° 3, 4, 6, 8, 9, 10.

Puisque ces exigences 3, 4, 6 ont les mêmes type d'exigence (Indexes), manuel (CML) et le nom de la fonction de recherche (Search Item by Number), le nom des Use Cases de ces exigences est le même : **UC_Search_CML_Search Item by Number**. De même, le Use Case commun des exigences 8, 9, 10 est **UC_Search_CML_Search Item by Product**.



La liste des exigences et des Use Cases devient :

N°	Chemin d'arborescence	Use Case		Type
1	IHM>Indexes>CML>		2	Mère
2	IHM>Indexes>CML>Search Item by Number>		3	Mère
3	IHM>Indexes>CML>Search Item by Number>GENERAL DESCRIPTION>	**UC_Search_CML_Search Item by Number**	4	Fille
4	IHM>Indexes>CML>Search Item by Number>FORM>	**UC_Search_CML_Search Item by Number**	4	Fille
5	IHM>Indexes>CML>Search Item by Number>RESULTS>		4	Mère
6	IHM>Indexes>CML>Search Item by Number>RESULTS>TARGETS>	**UC_Search_CML_Search Item by Number**	5	Fille
7	IHM>Indexes>CML>Search Item by Product>		3	Mère
8	IHM>Indexes>CML>Search Item by Product>GENERAL DESCRIPTION>	**UC_Search_CML_Search Item by Product**	4	Fille
9	IHM>Indexes>CML>Search Item by Product>FORM>	**UC_Search_CML_Search Item by Product**	4	Fille
10	IHM>Indexes>CML>Search Item by Product>RESULTS>	**UC_Search_CML_Search Item by Product**	4	Fille

Une fois toutes les exigences de type « fille » pure ont leurs Use Cases, l'affectation des Use Cases aux exigences « mères » est commencée. L'outil va maintenant balayer ces exigences du plus bas niveau hiérarchique au plus haut.

L'outil commence avec l'exigence numéro 5, une exigence « mère » ayant niveau hiérarchique 4. Sa fille est l'exigence numéro 6, qui a déjà son Use Case **UC_Search_CML_Search Item by Number**. D'après la règle de génération 2a, cette exigence est aussi couverte par ce Use Case. Donc le Use Case correspondant pour l'exigence 5 est **UC_Search_CML_Search Item by Product**.

N°	Chemin d'arborescence	Use Case		Type
1	IHM>Indexes>CML>		2	Mère
2	IHM>Indexes>CML>Search Item by Number>		3	Mère
3	IHM>Indexes>CML>Search Item by Number>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Number	4	Fille
4	IHM>Indexes>CML>Search Item by Number>FORM>	UC_Search_CML_Search Item by Number	4	Fille
5	IHM>Indexes>CML>Search Item by Number>RESULTS>	**UC_Search_CML_Search Item by Number**	4	Mère
6	IHM>Indexes>CML>Search Item by Number>RESULTS>TARGETS>	UC_Search_CML_Search Item by Number	5	Fille
7	IHM>Indexes>CML>Search Item by Product>		3	Mère
8	IHM>Indexes>CML>Search Item by Product>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Product	4	Fille
9	IHM>Indexes>CML>Search Item by Product>FORM>	UC_Search_CML_Search Item by Product	4	Fille
10	IHM>Indexes>CML>Search Item by Product>RESULTS>	UC_Search_CML_Search Item by Product	4	Fille

L'outil continue avec les exigences « mères » du niveau plus haut, ici sont les exigences 2 et 7 du niveau 3. Comme toutes les exigences filles 3, 4, 5, 6 de l'exigence 2 sont couverte par le même Use Case **UC_Search_CML_Search Item by Number**, donc elle est aussi couverte par ce Use Case en appliquant la règle 2a de génération des Use Cases. De la même façon, on peut déduire le Use Case **UC_Search_CML_Search Item by Product** pour l'exigence 7.

N°	Chemin d'arborescence	Use Case		Type
1	IHM>Indexes>CML>		2	Mère
2	IHM>Indexes>CML>Search Item by Number>	**UC_Search_CML_Search Item by Number**	3	Mère
3	IHM>Indexes>CML>Search Item by Number>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Number	4	Fille
4	IHM>Indexes>CML>Search Item by Number>FORM>	UC_Search_CML_Search Item by Number	4	Fille
5	IHM>Indexes>CML>Search Item by Number>RESULTS>	UC_Search_CML_Search Item by Number	4	Mère
6	IHM>Indexes>CML>Search Item by Number>RESULTS>TARGETS>	UC_Search_CML_Search Item by Number	5	Fille
7	IHM>Indexes>CML>Search Item by Product>	**UC_Search_CML_Search Item by Product**	3	Mère
8	IHM>Indexes>CML>Search Item by Product>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Product	4	Fille
9	IHM>Indexes>CML>Search Item by Product>FORM>	UC_Search_CML_Search Item by Product	4	Fille
10	IHM>Indexes>CML>Search Item by Product>RESULTS>	UC_Search_CML_Search Item by Product	4	Fille



Il nous reste que l'exigence 1. On sait que toutes ses exigences sont couverte par les deux **Use Cases UC_Search_CML_Search Item by Number** et **UC_Search_CML_Search Item by Product**, donc d'après la règle 2b, cette exigence est aussi couverte, mais on ne peut pas l'affecter un Use Case particulier. Donc pour ce cas, l'outil va marquer « Covered » pour cette exigence.

La liste des exigences après la génération des Use Cases est :

N°	Chemin d'arborescence	Use Case		Type
1	IHM>Indexes>CML>	**Covered**	2	Mère
2	IHM>Indexes>CML>Search Item by Number>	UC_Search_CML_Search Item by Number	3	Mère
3	IHM>Indexes>CML>Search Item by Number>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Number	4	Fille
4	IHM>Indexes>CML>Search Item by Number>FORM>	UC_Search_CML_Search Item by Number	4	Fille
5	IHM>Indexes>CML>Search Item by Number>RESULTS>	UC_Search_CML_Search Item by Number	4	Mère
6	IHM>Indexes>CML>Search Item by Number>RESULTS>TARGETS>	UC_Search_CML_Search Item by Number	5	Fille
7	IHM>Indexes>CML>Search Item by Product>	UC_Search_CML_Search Item by Product	3	Mère
8	IHM>Indexes>CML>Search Item by Product>GENERAL DESCRIPTION>	UC_Search_CML_Search Item by Product	4	Fille
9	IHM>Indexes>CML>Search Item by Product>FORM>	UC_Search_CML_Search Item by Product	4	Fille
10	IHM>Indexes>CML>Search Item by Product>RESULTS>	UC_Search_CML_Search Item by Product	4	Fille

D'où toutes les exigences sont couvertes par les Use Cases (ratio 100%).

EXPORTATION DES FICHIERS LOG

Des fichiers log sont des fichiers contenant les résultats du processus : la matrice de traçabilité et le rapport sur la couverture des Use Cases générés.

La matrice de traçabilité est représentée par un fichier Excel. Les exigences dans ce fichier sont représentées ligne par ligne avec les attributs : Identifiant d'exigence, Standard, Origine, les Traces (si existe), Nom d'exigence, Numéro hiérarchique, ParentId, Identifiant d'exigence mère, Chemin d'arborescence, Niveau hiérarchique, Chemin d'arborescence complet et le Use Case correspondant. Ce fichier Excel sera créé automatiquement dans le même répertoire en portant le même nom du fichier Word, si un fichier ayant le même nom existe déjà, il sera remplacé par le nouveau fichier. Une autre solution proposée est que le log sera crée sous forme d'une feuille incluse dans le fichier Excel de traçabilité.

Le rapport du processus peut être soit un fichier particulier exporté durant le processus soit inclus dans la matrice de traçabilité en l'ajoutant à la fin du tableau. Les informations contenues dans le rapport sont :

- Le nombre total d'exigences de la matrice de traçabilité.
- Le nombre des exigences couvertes par les Use Cases générés.
- Le ratio de la couverture (en pourcentage) : est calculé en divisant le nombre des exigences couvertes par le nombre total des exigences. Ce ratio est le résultat principal du processus, il permet d'estimer la performance du processus ainsi que les règles de génération. Les règles de générations doivent donner un ratio de plus de 80% pour n'importe quel projet.
- Le nombre d'exigences nulles (les exigences non livrés).
- Le nombre de Use Cases générés.
- Le temps de l'exécution du processus (en format hh :mm :ss).

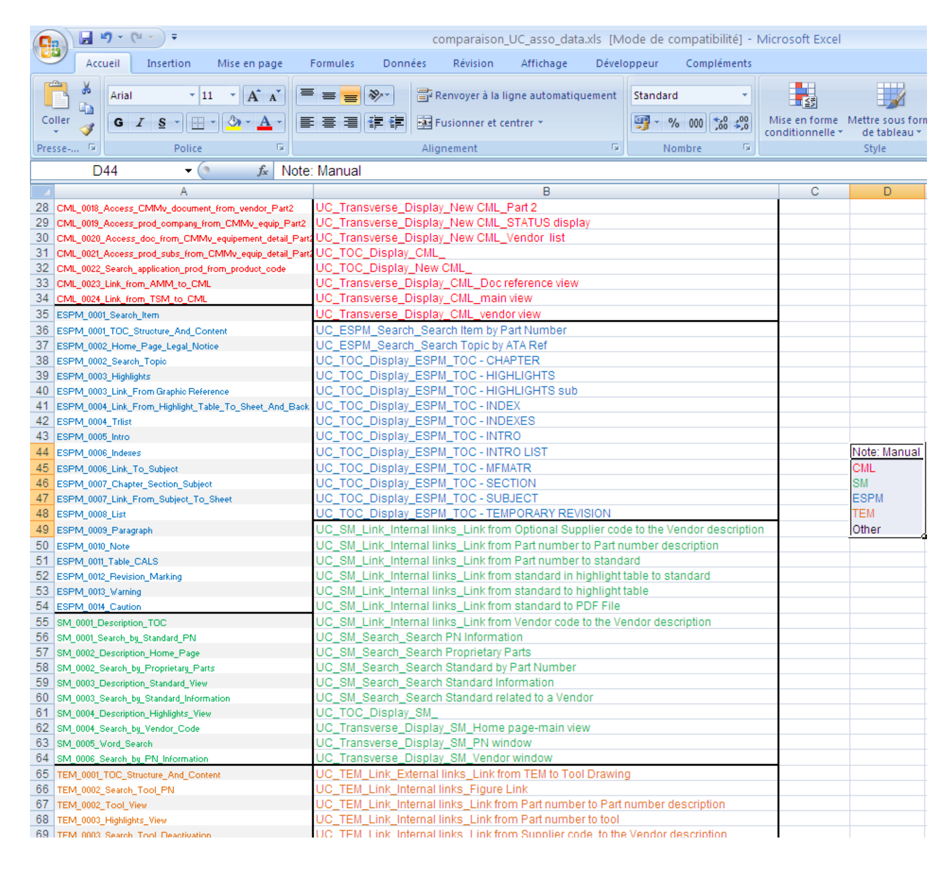


Exemple : La partie résultat du projet Associated Data V3 :

- Total number of requirements = 332
- Number of covered requirements = 269
- Ratio = 81,02%
- Requirement null : 2
- Number of UC generated = 116
- Running time = 00:02:07
- List of UC generated

COMPARAISON AVEC LES USE CASES DANS HPQC

La liste des Use Cases générés par le processus est maintenant utilisée pour effectuer la comparaison avec les Use Cases existants du produit dans HPQC. La comparaison est effectuée par manuel. En comparant avec les Use Cases générés par le processus qui assurent la couverture totale sur les exigences du BRD, les Use Cases existant dans HPQC seront modifiés afin de compléter la traçabilité et la couverture. Cette action est faite pour les projets existants dans HPQC, c'est-à-dire les projets qui ont déjà les Use Cases de validation créés par le chef de produit.


comparaison_UC_asso_data.xls [Mode de compatibilité] - Microsoft Excel

D44 Note: Manual

	A	B	C	D
28	CML_0018_Access_CMMv_document_from_vendor_Part2	UC_Transverse_Display_New CML_Part 2		
29	CML_0019_Access_prod_company_from_CMMv_equip_Part2	UC_Transverse_Display_New CML_STATUS display		
30	CML_0020_Access_doc_from_CMMv_equipement_detail_Part2	UC_Transverse_Display_New CML_Vendor list		
31	CML_0021_Access_prod_subs_from_CMMv_equip_detail_Part2	UC_TOC_Display_CML_		
32	CML_0022_Search_application_prod_from_product_code	UC_TOC_Display_New CML_		
33	CML_0023_Link_from_AMM_to_CML	UC_Transverse_Display_CML_Doc reference view		
34	CML_0024_Link_from_TSM_to_CML	UC_Transverse_Display_CML_main view		
35	ESPM_0001_Search_Item	UC_Transverse_Display_CML_vendor view		
36	ESPM_0001_TOC_Structure_And_Content	UC_ESPM_Search_Search Item by Part Number		
37	ESPM_0002_Home_Page_Legal_Notice	UC_ESPM_Search_Search Topic by ATA Ref		
38	ESPM_0002_Search_Topic	UC_TOC_Display_ESPM_TOC - CHAPTER		
39	ESPM_0003_Highlights	UC_TOC_Display_ESPM_TOC - HIGHLIGHTS		
40	ESPM_0003_Link_From Graphic Reference	UC_TOC_Display_ESPM_TOC - HIGHLIGHTS sub		
41	ESPM_0004_Link_From_Highlight_Table_To_Sheet_And_Back	UC_TOC_Display_ESPM_TOC - INDEX		
42	ESPM_0004_Trlist	UC_TOC_Display_ESPM_TOC - INDEXES		
43	ESPM_0005_Intro	UC_TOC_Display_ESPM_TOC - INTRO		
44	ESPM_0006_Indexes	UC_TOC_Display_ESPM_TOC - INTRO LIST		Note: Manual
45	ESPM_0006_Link_To_Subject	UC_TOC_Display_ESPM_TOC - MFMATR		CML
46	ESPM_0007_Chapter_Section_Subject	UC_TOC_Display_ESPM_TOC - SECTION		SM
47	ESPM_0007_Link_From_Subject_To_Sheet	UC_TOC_Display_ESPM_TOC - SUBJECT		ESPM
48	ESPM_0008_List	UC_TOC_Display_ESPM_TOC - TEMPORARY REVISION		TEM
49	ESPM_0009_Paragraph	UC_SM_Link_Internal links_Link from Optional Supplier code to the Vendor description		Other
50	ESPM_0010_Note	UC_SM_Link_Internal links_Link from Part number to Part number description		
51	ESPM_0011_Table_CALS	UC_SM_Link_Internal links_Link from Part number to standard		
52	ESPM_0012_Revision_Marking	UC_SM_Link_Internal links_Link from standard in highlight table to standard		
53	ESPM_0013_Warning	UC_SM_Link_Internal links_Link from standard to highlight table		
54	ESPM_0014_Caution	UC_SM_Link_Internal links_Link from standard to PDF File		
55	SM_0001_Description_TOC	UC_SM_Link_Internal links_Link from Vendor code to the Vendor description		
56	SM_0001_Search_by_Standard_PN	UC_SM_Search_Search PN Information		
57	SM_0002_Description_Home_Page	UC_SM_Search_Search Proprietary Parts		
58	SM_0002_Search_by_Proprietary_Parts	UC_SM_Search_Search Standard by Part Number		
59	SM_0003_Description_Standard_View	UC_SM_Search_Search Standard Information		
60	SM_0003_Search_by_Standard_Information	UC_SM_Search_Search Standard related to a Vendor		
61	SM_0004_Description_Highlights_View	UC_TOC_Display_SM_		
62	SM_0004_Search_by_Vendor_Code	UC_Transverse_Display_SM_Home page-main view		
63	SM_0005_Word_Search	UC_Transverse_Display_SM_PN window		
64	SM_0006_Search_by_PN_Information	UC_Transverse_Display_SM_Vendor window		
65	TEM_0001_TOC_Structure_And_Content	UC_TEM_Link_External links_Link from TEM to Tool Drawing		
66	TEM_0002_Search_Tool_PN	UC_TEM_Link_Internal links_Figure Link		
67	TEM_0002_Tool_View	UC_TEM_Link_Internal links_Link from Part number to Part number description		
68	TEM_0003_Highlights_View	UC_TEM_Link_Internal links_Link from Part number to tool		
69	TEM_0003_Search_Tool_Deactivation	UC_TEM_Link_Internal links_Link from Supplier code to the Vendor description		

Figure 28 - Fichier de comparaison des Use Cases du produit Associated Data V3

Pour les nouveaux projets, c'est-à-dire les projets qui n'ont pas encore de Use Case de validation dans HPQC, la liste des noms des Use Cases sera utilisée pour importer dans la partie « Requirements » de HPQC. Avec les noms des Use Cases importé dans le produit, le chef de produit peut maintenant commencer à initialiser la génération des Use Cases en rédigeant le contenu de chaque Use Case, ce qui assure déjà la traçabilité et la couverture totale avec les exigences.



CONCLUSION

DIFFICULTÉS RENCONTRÉS

Durant la réalisation du sujet de stage, j'ai rencontré quelques difficultés. La première vient de mon manque de connaissance sur les sujets comme la méthode de V&V et aussi les projets informatiques de SEVS. Comme je ne les ai jamais vu à la faculté, au début, j'avais l'air tout à fait dépaysé. Ensuite, la prise de connaissance sur les projets de SEVS avec les éléments spécialisés (surtout en aéronautique) ainsi que la fonctionnalité des outils AirN@v et ses documents en SGML inclus, m'a demandé un certains temps. En réalité, le temps que j'ai passé sur ces étapes de prise de connaissance et d'analyse sur les projets a un peu dépassé la limite que j'avais prévue.

J'ai du encore rencontré des difficultés dans la gestion du temps de l'étape de test. Puisque les tests effectués sur les macros SBA sont plus long que prévu (les raisons ont été détaillées dans la partie précédente), le planning a du être modifié avec un décalage des étapes qui suivent celle de test (l'analyse du 3e scénario, rédaction d'un guide d'utilisation…). Ces difficultés m'ont montré l'importance de prévoir les échéances ou les délais possibles au lieu de construire un plan trop rigide.

Une autre difficulté vient de l'indisponibilité de mon maître de stage. Dû aux fonctions de SEVS, les membres en étant chef de produit doivent avoir beaucoup de réunion avec les différents acteurs (business, sous-traitance, les autres services au sein de SE…), c'est pourquoi mon maître de stage n'était pas toujours à son bureau pour que je puisse lui poser les questions quand j'avais des soucis. Nous avons géré ce problème par les petites conversations qui sont faites souvent au début de chaque journée, de 8h à 9h. De plus, la messagerie Outlook fourni par Airbus m'a beaucoup aidé pour les contacts avec mon maître de stage n'importe quand, j'ai pu aussi prendre connaissance sur les politiques de communication dans une grande entreprise.

Ces difficultés ne m'ont pas beaucoup embarrassé, par contre, en réfléchissant sur l'origine de ces difficultés et leurs solutions, j'ai pu mieux apprendre et améliorer le savoir-faire et les façons de gérer un projet informatique en réalité, ce que je n'avais pas pu voir complètement à l'université.

LIMITES ET ÉVOLUTIONS DE L'OUTIL

Cet outil de représentation dynamique est une évolution à l'ensemble des fonctions de gestion dé projets en service. Néanmoins, elle présente quelques limites et certaines évolutions pourraient être réalisées.

Les limites de l'outil sont déjà détaillées dans la partie d'analyse des scénarios et des langages de codage. Ici je propose juste quelques chemins d'évolution de cet outil que je n'ai pas le temps de suivre et de développer pour le rendre plus performant et plus efficace dans le futur.

Continuation avec les Test Cases : la partie d'impact de l'outil avec HPQC, plus précisément avec les Test Cases, n'est pas étudiée profondément dans le sujet de stage. En fait, l'outil peut encore avancer dans le processus de la méthode de V&V en ajoutant les procédures qui génère, pas seulement la traçabilité entre les Use Cases et les exigences mais aussi entre les Use Cases et leurs Test Use Cases. Les Test Cases d'un Use Case seront générés en façon automatique par l'outil, classés dans l'ordre des étapes de Test et correspondront à l'exigence couverte par ce Use Case. Pour réaliser cette procédure, il faudra ajouter les règles de génération pour les Test Cases et les méthodes pour pouvoir mettre à jour les projets dans HPQC.

Améliorer la performance de l'outil : chercher à trouver une méthode pour réduire le temps de transport automatique les données du fichier Word au fichier Excel (car pour le moment, cette procédure prend encore



la plupart du temps du processus), ajouter encore les règles de générations des Use Cases pour que l'outil puisse s'adapter à tous les projets de SEVS, puisque pour le moment, les règles de l'outil ne peuvent s'appliquer que sur les projets de type AirN@v, c'est-à-dire consultation électronique de documents..

Perfectionner l'utilisation de l'outil : faciliter l'installation et l'utilisation de l'outil, perfectionner le code pour rendre l'outil plus « libre » à utiliser (plus de fonctionnalité ou plus d'option d'exécution de l'outil à choisir par l'utilisateur). Ajouter et améliorer l'interface homme-machine pour simplifier autant que possible l'utilisation de l'outil. Effectuer les tests plus précis et mettre à jour l'outil pour réduire les erreurs et les défauts de l'outil. Mettre à jour le guide d'utilisation…

Améliorer la fonctionnalité de l'outil : ajouter les fonctions de filtre, par exemple, sur les attributs du BRD, les fonctions de recherche des exigences basées sur leurs attributs ainsi que la capacité de travailler sur plusieurs BRD de plusieurs projets…

Tous ces propositions pourront être élaborées dans les prochaines versions de l'outil vu que dans mon stage, je n'ai pas de temps de les aborder.



RÉFLEXION SUR LE SYSTÈME D'INFORMATION

LA GESTION DES PROJETS INFORMATIQUES D'AIRBUS – GPP TOOLKIT

Les projets informatiques gérés par le Customer Service (S) incluant SEVS sont les projets sur les produits qui mettent à disposition des clients la documentation technique permettant d'assurer la maintenance des avions en services. Le GPP Toolkit (Generic Project Process) est une méthodologie utilisé par le service pour la gestion de ces projets de système informatique d'Airbus. GPP est une modèle générique, adapté en fonction des caractéristiques spécifique de chaque domaine, technologie, programme ou projet. L'instruction de GPP introduit le cycle de vie, l'organisation et les règles appliquant sur les projets du système information.

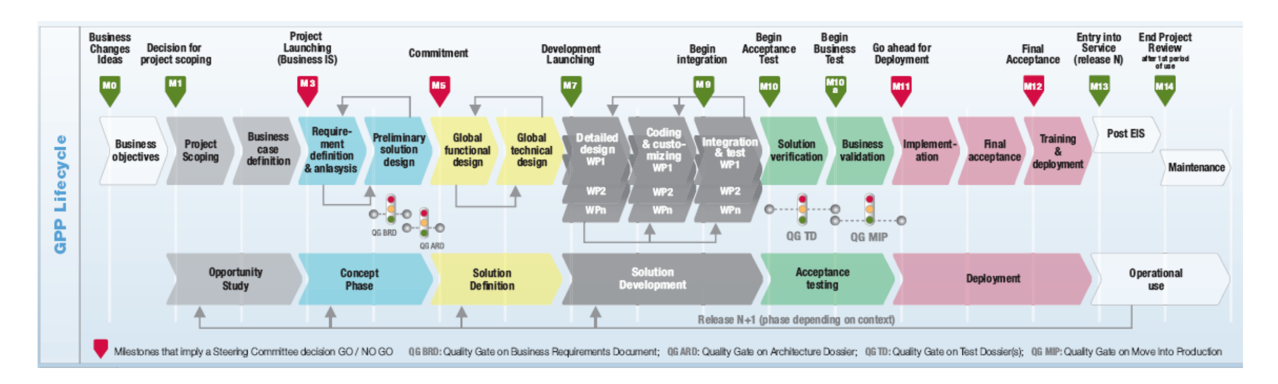


Figure 29 - Cycle de vie de GPP

Le cycle de vie de GPP a la forme qui se ressemble au cycle de vie en V qui se compose par des phases, des jalons et les portes de qualité.

- **Phase** : 7 phases, chacune peut être divisée en sous-phases. Chaque phase doit être accomplie dans l'ordre chronologique afin de s'assurer que ces principaux objectifs sont atteints avant de passer à la suivante. Les 7 phases principales de GPP dans l'ordre chronologique sont Etude de l'opportunité, Conception, Définition de la solution, Développement de la solution, Test d'acceptation, Déploiement, Utilisation opérationnelle.

- **Jalons** (**Milestones**) : 14 jalons pour marquer les points de contrôle importants : les jalons en rouge (M3, M5, M11, M12) sont les étapes de GO / NO GO qui impliquent une décision du comité de pilotage. Chaque jalon permet de faire une estimation sur l'état actuel du projet, d'assurer que les décisions clés relatives à l'avancement du projet sont effectuées, d'évaluer si le projet est prêt à avancer à la prochaine phase, le non atteint des objectifs peut conduire à re-travailler sur la phase actuelle ou même à abandonner du projet. Il y a 11 jalons principaux dont 4 GO / NO GO.

- **Etapes qualité (Quality Gates)** : pour l'évaluation des livrables clés. Ce processus est réalisé en 2 étapes. La première étape est la mise en place de l'accord du processus pour déterminer les fonctions des acteurs et les livrables impliqués, les critères d'acceptation et le chemin restant. La 2e étape est l'évaluation des livrables en se basant sur les critères d'acceptation. Les livrables peuvent être le BRD (Business Requirement Dossier), ARD (Architecture Dossier) avant l'étude de la spécification, TD (Test Dossier) pour fournir au Test, MIP (Move Into Production) avant le déploiement.



LE CHEF DE PROJET AU SEIN D'UN PROJET INFORMATIQUE

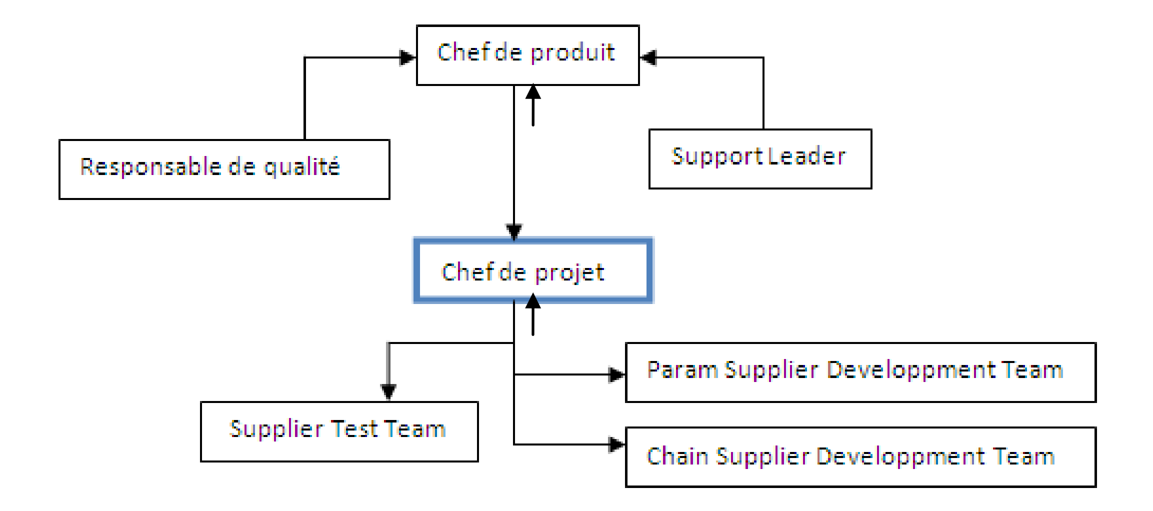


Figure 30 - Structure de l'organisation d'un projet

Le responsable des acteurs dans l'organisation du projet doit accomplir l'objectif. Le rôle (qui fait quoi) et la responsabilité (qui décide quoi) du projet est partagé par les différents acteurs : le chef de produit, le chef de projet, le responsable de qualité, le responsable support et les équipes de développement et de test. Ici on ne s'intéresse que au rôle du chef de projet.

Le chef de projet joue un rôle intermédiaire dans le projet. Il coordonne avec le chef de produit le suivi du développement du projet, d'autre part, il délègue et veille les phases de développement et de test technique réalisées par les sous-traitants.

TÂCHES

Le chef de produit, après avoir travaillé avec les clients, surtout les compagnies aériennes, sur leurs besoins et leurs références, va effectuer les études de l'opportunité et de la faisabilité du produit. Les exigences sont ensuite définies et rédigées dans le Business Requirement Dossier afin de livrer au chef de projet pour la phase suivante.

Le chef de projet, en recevant le BRD avec les autres documents nécessaires, va établir le plan de développement en définissant le calendrier associé et les jalons. Le calendrier doit contient le calendrier des « paquets » de travail (work package) décrit dans le projet. Ce calendrier doit se baser sur les contraintes et la dépendance des tâches. Une fois établi, le calendrier sera proposé à l'équipe de développement et de test et sera revu entre eux. Chez Airbus, le calendrier est géré par l'application PSNext. Dans le cas de mise à jour du calendrier à cause d'une modification fonctionnelle ou d'une contrainte de développement, une réunion avec le chef de produit est nécessaire pour la décision finale.

Ensuite, il doit définir la solution technique, c'est-à-dire la spécification du produit, en rédigeant le livrable ARD (Architecture and Design) et en s'assurant que la spécification couvre toutes les exigences. Le budget et les contrats avec les équipes de développement seront effectués dès la livraison du ARD. Le budget est géré par l'application PSNext et se base sur les contrats avec les sous-traitants. Chaque mois, le chef de projet doit faire une revue avec le chef de produit sur le reste du budget dans la réunion d'étape.

Durant la phase de développement et la phase de test, le chef de projet doit souvent effectuer les réunions entre les équipes de sous-traitants pour estimer l'état actuel du projet et les réunions avec le chef de produit



pour lui fournir ces informations. Il doit aussi contrôler le projet pour assurer la livraison au temps opportun, la qualité et le budget.

A la fin, le chef de projet doit gérer la campagne de vérification du produit pour assurer la pré-intégration du produit et son intégration dans le portail pour fournir à l'étape de validation et de mise en service du produit.

LIVRABLES

La liste des livrables sous responsabilité du chef de projet sont différents selon les étapes. Ils sont rédigés et sont validés à la fin de chaque phase sous forme des jalons. Voici les livrables principales produits durant le processus :

- Documents de gestion du projet (livrés à la Révision de la Définition du Projet) :
    - Plan du projet
    - Plan de vérification et de validation
    - Planning et budget
    - Plan de gestion de configuration du projet
- Documents du projet (livrés avant et durant la phase de développement et de test des sous-traitants)
    - Dossier de spécification fonctionnelle
    - Dossier d'architecture
    - Interfaces
    - Dossier de Test
    - Matrice de traçabilité entre exigences et spécifications
    - Matrice de traçabilité entre spécifications et tests
    - Dossier de conception
    - Catalogue de configuration
    - Note de livraison
- Documents d'intégration dans le portail
    - Liste des exigences d'intégration dans le portail
- Documents d'intégration informatique (fournir à la phase de Test)
    - Dossier de test d'application informatique
    - Questionnaire de test d'application
    - Manuel d'installation on-board
    - Liste des exigences d'intégration on-board



RÉUNIONS

Comme le rôle du chef de projet est considéré comme le cœur du projet, il doit participer aux différentes réunions durant le processus avec les différents acteurs.

La réunion d'étape du projet (**Project Progress Meeting**) est faite une fois par mois, entre le chef de produit, le chef de support et le chef de projet pour fournir une visibilité sur le développement au chef de produit et les indicateurs de suivis.

La réunion de développement du projet (**Project Development Meeting**) est une réunion avec l'équipe de développement. Cette réunion sert à fournir les informations au chef de projet sur le développement du projet.

La réunion de revue du code du produit (**Project Code review**), en particulier, est faite avant chaque revue de réception du fournisseur. Elle se passe entre le chef de projet et l'équipe de développement.

Durant la phase de test, chaque semaine, une réunion de test (**Project Test Meeting**) est passée avec l'équipe de test pour fournir une visibilité sur la phase de vérification du produit.

3 réunions de validation de la spécification (**Proofreading SD Review**) doivent être passées avec le chef de produit avant la Révision de spécification afin de corriger et valider la spécification.

Il y a aussi certains réunions non périodique comme la réunion de qualité (**Quality Meeting**) faite avec le responsable de qualité sur demande ou la réunion pour les changements du contour fonctionnel (**Project CCB**) de la version du produit avec la participation de chef de produit si nécessaire.



BILAN PERSONNEL

PLAN TECHNIQUE

En réalisant le sujet de stage, je me suis rendu compte en réalité que les langages informatiques que j'ai appris à l'université (Java, Oracle, SQL, C++…) ne sont pas nécessaire pour la réalisation du stage. De plus, la méthode de gestion des projets informatique utilisée à SEVS est le cycle de vie en V, une méthode un peu ancienne et moins utilisée aujourd'hui… Cela m'a un peu étonné dans les premiers jours de stage, quand j'ai du prendre connaissance sur l'existant du sujet de stage.

Pourtant, l'apport le plus important d'un stage n'est pas d'appliquer des langages de codage ou des méthodes informatiques déjà appris à l'université dans le stage mais d'acquérir, au travers du sujet de stage, un **savoir-faire** sur la mise en œuvre de techniques informatiques dans un environnement professionnel particulier. Les connaissances de la théorie ou la pratique qu'on obtenait à l'université ne sont jamais suffisantes pour la vie professionnelle, ce qu'on doit savoir est la façon de se « débrouiller » avec les projets informatiques en apprenant à s'adapter avec n'importe quel méthode d'analyse et de développement.

Durant la réalisation du sujet de stage, j'ai du prendre en compte des connaissances sur les langages que je n'ai jamais vu avant : le Visual Basic for Applications (VBA), le Batch Code et aussi un peu de code sur la génération d'un fichier par Document Factory… Mes connaissances sur les langages de codage de base comme Java, C++ m'ont aidé à comprendre plus facilement les principes du VBA, ainsi que le Batch Code qui est proche des commandes MS-DOS et aussi des commandes de shell d'UNIX. C'est pourquoi, ma formation sur les méthodes de développement n'a pas pris beaucoup de temps.

Par contre, mon adaptation avec la méthodologie V&V n'a pas été tout à fait simple et rapide comme les langages de développement. Cette méthodologie est totalement différente des méthodes d'analyse que j'ai apprises. De plus, le manque de documentation sur cette méthodologie me gênait aussi dans la prise de connaissance. Grâce aux explications avec enthousiasme de mon maître de stage Jean BOUSQUET et ma collègues Nathalie SIGNES, j'ai pu finalement bien avancer dans mes projets avec cette méthode, comprendre sa fonctionnalité et ses points forts pour lesquels elle est utilisée dans la gestion des projets informatiques d'Airbus.

J'ai aussi pu acquérir les expériences dans la planification de la réalisation du projet. J'ai du choisir une méthode avec des étapes que j'ai détaillé dans la partie de réalisation du projet, qui à la fois ressemble au courbe du soleil et au cycle de vie en V. J'ai aussi appris à construire un planning flexible en utilisant la méthode de Gant avec les cas échéants, les tâches que je pouvais faire simultanément pour augmenter la productivité et réduire le temps de réalisation au lieu d'un planning trop rigide, qui pourrait être totalement bloqué si une étape n'est pas effectué comme prévu (retard ou annulation).

En tous cas, cette période en entreprise m'a donné l'occasion de réutiliser une partie de mes enseignements reçus à l'université ainsi d'approfondir mes connaissances (savoir appliquer les méthodes d'analyse et de gestion de projet sur un projet réel, pour la conduite de projet et aussi la planification), j'ai pu aussi apprendre des nouvelles méthodes d'analyse et de développement, surtout la façon pour s'adapter avec n'importe quel langage de codage. Ce sont les choses que sans ce stage, je n'aurai pas pu savoir.



PLAN PROFESSIONNEL

L'apport du stage n'est pas seulement du côté technique, ce stage m'a permis de voir comment une entreprise fonctionne dans la réalité. C'est la raison la plus importante pour laquelle j'avais décidé de choisir la MIAGe qui oblige les étudiants à effectuer un stage à la 3e année, la vie professionnelle d'un employé dans l'entreprise m'intéressant beaucoup depuis longtemps. En réaliser un stage dans une grand entreprise comme Airbus, j'ai pu l'observer et partager l'ambiance professionnelle, ce qui me plaisait beaucoup.

Mon bureau était dans la même salle de tous les 7 membres de SEVS. C'est pourquoi je pouvais observer les façons de communication interpersonnelles, savoir comment les chefs de produit pouvais gérer leurs projets, leurs tâches et leurs emplois du temps avec les réunions, les formations et les heures de travail individuel. L'ambiance de travail était bonne. Mes collègues m'ont beaucoup aidé dans mon stage en expliquant les choses que je n'avais pas compris, me donnant les conseils… même si, à cause des réunions, ils n'avaient pas beaucoup de temps à me consacrer.

Du côté de la relation avec mon tuteur, puisque mon tuteur n'est pas toujours disponible à son bureau, il doit participer à des réunions, c'est pourquoi on a décidé de faire un point au début de la journée de travail, avant 9h, pour que je lui pose mes questions ou qu'il m'explique ce que je devrais réaliser… Chaque 2 ou 3 jours, à la fin de la journée, j'ai pris l'habitude de lui envoyer un mail en présentant l'état de mon travail, les résultats d'une étape pour sa validation… Grâce à cette habitude, je pouvais bien gérer mes échéances de mon planning, prévoir les étapes suivantes, faire des rapports…

Durant le stage, j'ai pu aussi participer à des réunions de l'équipe de SEVS, surtout dans la première semaine, pour m'adapter à l'ambiance du travail et mieux comprendre les projets. Il y a plusieurs types de réunions. Des réunions hebdomadaires entre les membres de SEVS, entre SEVS avec l'équipe de SBA permettant d'annoncer et informer sur l'état des projets ou des nouvelles, discuter pour trouver la solution pour un problème commun. Des réunions entre un membre de SEVS (chef de produit) avec les sous-traitants qui travaillent sur son projet pour la validation des étapes. Des réunions avec le côté business pour recueillir les exigences afin de construire ou modifier le BRD pour les prochaines versions d'application... La règle pour prendre la parole des réunions est souvent de faire tour de table, chacun présente sa partie dans un certains temps et les autres lui posent des questions si nécessaire. Parfois les débats passionnés sont soulevés. J'ai aussi eu l'occasion de présenter l'état de mon stage et les résultats de cette partie dans une réunion de SEVS et SBA. Avec les conseils très utiles de mes collègues autour de cette présentation, j'ai pu apprendre beaucoup sur la construction d'un plan de présentation avec l'ordre des parties indispensables, la façon pour attirer l'attention du public, pour bien expliquer pour le faire comprendre, pour argumenter mes opinions, et aussi les petits points comme l'attitude, la voix, les mots à utiliser… en appliquant mes connaissances sur les cours de communication de la MIAGe.

PLAN PERSONNEL

Mon sujet de stage est intéressant, même si au début il me semblait un peu abstrait car il n'est pas un sujet pur développement informatique. L'objectif du projet de stage demande surtout des réflexions pour proposer une façon, une méthode et des règles pour augmenter l'efficacité d'un processus. C'est pourquoi dans la réalisation de ce projet, les étapes d'analyse et d'étude des projets pour trouver les règles ont occupé la plupart du temps, plus que le temps pour la spécification, le développement et les tests. C'est aussi la caractéristique de ce métier, un chef de produit, qui ne réalise pas beaucoup de développement ou des tests mais surtout d'analyse des exigences, de rédaction du BRD, d'organisation et de gestion les campagnes de test de validation… des travaux plutôt du côté de gestion des projets informatiques, vers quoi la MIAGe s'oriente. Donc ce sujet de



stage et ce métier me semble parfaitement convenables. En réalité, grâce à ce stage, j'ai pu beaucoup avancer dans la suite de mes études pour devenir un chef de projet.



PLANNIFICATION DE STAGE

Mois	avr-11		mai-11				juin-11					juil-11				août-11				##
Semaine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Prise de connaissance sur les nouvelles méthodologies de gestion	■																			
Familiarisation avec les outils Airbus	■																			
Etude sur l'objectif du projet - Proposition de la solution		■																		
Analyse du projet AirN@v Workshop V3			■																	
Analyse du projet AirN@v Maintenance A350			■	■																
Analyse du projet AirN@v Associated Data V3				■	■															
Fini du cahier des charges					■															
Développement suivant la solution 1						■	■													
Test de la solution 1								■												
Développement suivant la solution 2									■											
Test de la solution 2										■	■	■								
Présentation de l'outil au service												■								
Rédiger les documentaires supports												■	■	■	■	■	■			
Développement suivant la solution 3 - Fini la version 1.0 de l'outil															■	■				
Test de la solution 3																	■			
Développer la version 1.1 de l'outil																	■	■	■	
Livraison de la version 1.1																				■
Rédiger du rapport de stage												■	■	■	■	■	■	■	■	■

BIBLIOGRAPHIE

Airbus People pour toutes les informations d'Airbus
http://airbuspeople.airbus.corp/employee_portal/

Différents cycles de vie par Anne-Marie Hugues
http://users.polytech.unice.fr/~hugues/GL/chapitre2.pdf

Traceability from Use Cases to Test Cases par Peter Zielczynski
http://www.ibm.com/developerworks/rational/library/04/r-3217/

V&V Lifecycle Methodologies par David F. Rico
http://davidfrico.com/vvbrief.pdf

Ingénerie des exigences de Borland CaliberRM
http://www.borland.com/fr/products/caliber/index.html

HPQC Tutorial
http://shrivv.com/Technicle/QA_HP_Tutorial.pdf

Les sites pour la formation en VBA et Batch Code
http://www.developpez.net
http://www.excel-downloads.com
http://forum.hardware.fr

Wikipédia, Encyclopédie libre
http://fr.wikipedia.org

Google traduction
http://translate.google.fr

GLOSSAIRES

AJCP	Adoc Job Card Publisher
BRD	Business Requirements Dossier (Le Cahier des Spécifications ou cahier des charges).
CaliberRM	une plate-forme de gestion des exigences logicielles
CASA	Construcciones Aeronáuticas Sociedad Anónima, une entreprise espagnole de construction d'avions créée le 9 mars 1923 à Séville, intégrée aujourd'hui à EADS.
CML	Consumable Material List (liste de matériel consumable).
CMMM	Component Maintenance Manual (manuel de maintenance des composants)
DFPRM	Duct and Fuel Pipe Repair Manual (manuel de réparation des conduits et des tuyaux de carburant)
EADS	European Aeronautic Defence and Space, société mère d'Airbus.
ESPM	Electrical Standard Practices Manual (manuel de pratiques standards électrique).
GIE	Groupement d'intérêt économique.
HPQC	Helwett-Packard Quality Center, un outil web de gestion des Tests.
IHM	Interface Humain-Machine.
IUC	Intended Use Case.
LATC	Live Animal Transportation Calculation pour A380 :un outil de calcul de la charge maximum des animaux, de confinement lors du transport des animaux vivants.
LETD	List of Effective Technical Data : un outil qui fournit aux clients toutes les informations sur l'expédition.
M&E	Maintenance & Ingeneering.
MOA	Maîtrise d'Ouvrage, le donneur d'ordre au profit de qui l'ouvrage est réalisé
MOE	Maîtrise d'œuvre, la personne chargée de la conduite opérationnelle de travaux.
MRTT	Multi Role Tanker/Transport (ravitailleurs multirôles).
NTM	Nondestructive Testing Manual (manuel de test non destructif).
OLCM	On Line Content Management
P/N	Part Number (d'un consommable, un outil ou un équipement standard de l'avion).



PACAT Parameters Catalogue

SAS Société par Actions simplifiée.

SBA Aircraft integrated platform support & services, un service d'Airbus.

SEVS Sigle du serivce Maintenance & Engineering Product Definition d'Airbus, le service dans lequel le stage est effectué.

SGML Standard Generalized Markup Language (langage normalisé de balisage généralisé) est un langage de description à balises.

SM Standards Manual (manuel standard).

SRM Structural Repair Manual (manuel de réparation structurelle).

TDSF Merge Tool Technical Data Service Merge Tool, outil utilisé pour fusionner les données du contrat de SAP avec des données de l'annuaire du portail d'EADS et l'information du mot de passe d'AirN@v pour les applications clientèles.

TEM Tool & Equipment Manual (manuel d'outil et d'équipement).

UC Use Case de validation (cas d'utilisation).

V&V method La méthodologie de Validation et Vérification.

Vx.y Version x.y (par exemple AirN@v Engineering V3.1).

VBA Visual Basic for Applications, une implémentation de Microsoft Visual Basic incluse dans toutes les applications de Microsoft Office.

WXB Wide Extra Body, une famille d'avions d'Airbus.